miro

-**https://miro.com/app/board/uXjVLhxZQmA=/**

github

-https://github.com/2110215-ProgMeth/cedt-project-2024-2-riatgome

Song of Twelve Feathers

โครงเรื่อง **(Story)** ตัวอย่าง

ในโลกแฟนตาซีที่เต็มไปด้วยอสูรและเวทมนตร์ มีตำนานเก่าแก่กล่าวไว้ว่า

> *“ผู้ใดถือครองขนนกศักดิ์สิทธิ์ทั้งสิบสอง จะสามารถเปิดประตูสู่พลังที่ยิ่งใหญ่ เกินหยั่งถึง…”*

แม้กาลเวลาจะผ่านไปเนิ่นนาน แต่คำทำนายนี้ยังคงดึงดูดผู้กล้าและเหล่าจอมเวทให้ออกผจญภัยหมายไขว่คว้า “ขนนก” ในตำนานให้ได้ ไม่ว่าจะเพื่อครอบครองพลัง หรือเพื่อกอบกู้โลกจากความมืดมิดที่คืบคลาน

คุณ รับบทเป็นนักผจญภัยพเนจร ผู้ครอบครอง “ขนนกแห่งการเริ่มต้น”
ซึ่งเป็นขนนกดอกแรกที่ว่ากันว่าเป็นกุญแจเปิดหนทางสู่การตามหาอีก 11 ขนนกที่เหลือ แต่ทุกเส้นทางย่อมเต็มไปด้วยอุปสรรค—มอนสเตอร์ดึกดำบรรพ์ โจรสลัดอวกาศ(?) ไปจนถึงปีศาจจากต่างมิติ (ขึ้นอยู่กับธีมและสไตล์ของด่าน)

ในการเดินทางผ่านด่านต่าง ๆ คุณจะต้องล้มบอสซึ่งเป็นผู้เฝ้าขนนกแต่ละดอกให้ได้ จึงจะได้ครอบครองขนนกนั้น
ทว่าโลกในเกมเป็นโลกที่ปรวนแปรเสมอ (ตามสไตล์โร๊คไลท์) ทุกครั้งที่คุณพลาดพลั้งล้มตาย
วิญญาณของคุณจะย้อนกลับมาพร้อมประสบการณ์บางส่วนที่ติดตัวไป และ “ขนนกแห่งการเริ่มต้น” ที่จะไม่มีวันสูญหาย
เพียงแต่ฉากและเส้นทางต่าง ๆ ก็อาจเปลี่ยนรูปแบบไป ทำให้แต่ละการผจญภัยในด่านเดิมไม่ซ้ำกัน

นอกจากนี้ ทุกครั้งที่คุณสามารถเอาชนะมอนสเตอร์หรือบอสในแต่ละด่านได้ คุณจะมีโอกาสได้รับ **รางวัล**
(ไอเทม/อุปกรณ์/เงิน/ค่าอัปเกรด) เพื่อเสริมพลังตัวละคร หรือปลดล็อกสกิลใหม่ ๆ ความสนุกจึงอยู่ที่การบริหารทรัพยากร
การวางกลยุทธ์เทิร์นเบสเพื่อปราบศัตรูในสถานการณ์ที่เปลี่ยนไปตลอดเวลา

โครงเรื่อง **(Story)** โดยสังเขป

ในโลกแฟนตาซีอันแปรผันและเต็มไปด้วยเวทมนตร์ มีตำนานบทหนึ่งที่ถูกขนานนามว่า **“Song of Twelve Feathers”**
ว่ากันว่าเป็นเพลงขับขานของขนนกศักดิ์สิทธิ์ทั้งสิบสอง ซึ่งกระจัดกระจายอยู่ตามดินแดนต่าง ๆ
ผู้ใดสามารถรวบรวมขนนกครบทั้งสิบสองดอกได้สำเร็จ จะได้รับพลังอำนาจเหนือกาลเวลา
พร้อมไขปริศนาแห่งการกำเนิดและการล่มสลายของโลกใบนี้

คุณ รับบทเป็นนักผจญภัยนิรนามที่ได้ครอบครอง “ขนนกแห่งการเริ่มต้น” โดยไม่รู้ที่มาที่ไป แต่ทันทีที่สัมผัสขนนกนั้น
คุณได้ยินเสียงเพลงประหลาด... บทเพลงอ่อนหวานแต่แฝงความลี้ลับ ราวกับร้องเรียกให้คุณออกเดินทางตามหา “ขนนก” ที่เหลือ
และกอบกู้โลกจากความมืดที่กำลังคืบคลาน

ทว่าทุกเส้นทางจะไม่เหมือนเดิมในแต่ละครั้งที่คุณออกผจญภัย (ตามสไตล์ **Roguelite**)
เพราะโลกจะเปลี่ยนไปเสมอเมื่อคุณพ่ายแพ้หรือกลับมาเริ่มต้นใหม่ แต่คุณยังคงเก็บเกี่ยวประสบการณ์และอัปเกรดบางส่วนไว้ได้เพื่อลุยต่อ

ใน **Song of Twelve Feathers** คุณจะได้เผชิญ

1. **การต่อสู้แบบเทิร์นเบส** ที่ต้องวางแผนการใช้สกิล เวทมนตร์ และไอเทมอย่างรอบคอบ
2. **ความท้าทายของโลกที่เปลี่ยนรูปแบบตลอดเวลา** ทำให้ต้องปรับกลยุทธ์ใหม่ในแต่ละด่าน
3. **บอสผู้เฝ้าขนนก** ดอกต่าง ๆ ที่มีพลังและลูกเล่นเฉพาะตัว
4. **ของรางวัล** สำหรับการพัฒนาตัวละคร ซึ่งคุณสามารถเลือกได้ตามสไตล์การเล่น ไม่ว่าจะเป็นไอเทมฟื้นพลัง
   อาวุธยุทโธปกรณ์ใหม่ ๆ หรือเงินทุนเพื่อปรับแต่งสกิล

บทเพลงแห่งขนนกจะขับขานเรื่องราวการต่อสู้และการเสียสละของคุณ จนกว่าคุณจะสามารถรวม “ขนนก” ทั้งสิบสองดอกได้ครบ
และเปิดเผยความจริงที่ซ่อนอยู่เบื้องหลังตำนานโบราณนี้

> **“เมื่อเสียงขับขานของขนนกสิบสองดังประสาน**
> **จะนำพาจิตวิญญาณไปสู่ราตรีแห่งการตื่นรู้...”**

จงออกผจญภัยและแต่งบทเพลงของคุณเองใน **Song of Twelve Feathers**!

บอสในแต่ละธีม

1) "อัคคีวิญญาณ" (ธีมภูเขาไฟ/ดินแดนเพลิง)

ชื่อบอส:

- **"อัคคีวิญญาณ" (Spirit of the Inferno)**
- หรือจะตั้งชื่อไทยให้เข้มขลัง เช่น "จ้าวอัคคีโบราณ"

ตำนาน/เนื้อเรื่องย่อ:

- ณ ใจกลางภูเขาไฟ "มรกตเพลิง" มีวิญญาณแห่งอัคคีอาศัยอยู่ในรูปกายของโกเลมยักษ์ที่หลอมรวมกับลาวา
- เชื่อกันว่าเป็นวิญญาณผู้พิทักษ์ขนนกดอกหนึ่งที่เรียกว่า **"ขนนกแห่งเปลวสุริยัน"** (Feather of Solar Flame)
- ความโกรธเกรี้ยวของมันทำให้ผืนพิภพแถบนั้นไหม้เกรียม ผู้คนจึงหวาดกลัวที่จะย่างกรายเข้าไปใกล้

รูปแบบ/คาแรกเตอร์:

- ตัวใหญ่เต็มพื้นที่ เวลาขยับจะมีลาวาหยดลงมาเป็นวงไฟบนพื้น
- มี "เปลวไฟ" ลุกโชนที่มือและหลัง ชี้ให้เห็นถึงพลังเพลิงอันแรงกล้า

สกิลหลัก (ตัวอย่าง):

1. **Eruption Smash** – ทุบพื้นอย่างรุนแรง สร้างความเสียหายธาตุไฟ (AoE) ให้กับผู้เล่น และอาจทำให้ติดสถานะ "เผาไหม้" (Burn)
2. **Molten Shield** – สร้างเกราะลาวารอบตัว ลดความเสียหายที่ได้รับในเทิร์นถัดไป และสะท้อนความเสียหายกลับเล็กน้อยหากผู้เล่นโจมตีระยะประชิด
3. **Flame Wave** – พ่นคลื่นไฟไปด้านหน้า โจมตีตัวละครเป็นเส้นตรง หากไม่ตั้งรับหรือไม่มีสกิลกันไฟ อาจโดนดาเมจอย่างรุนแรง
4. **Lava Surge (ท่าไม้ตาย)** – ร่ายพลังเรียกธารลาวาให้ไหลเข้ามาท่วมสนาม ลด HP ของผู้เล่นในทุกเทิร์นต่อเนื่องจนกว่าจะเอาชนะบอสหรือทำลายสนามลาวาได้ด้วยไอเทมหรือสกิลบางอย่าง

เงื่อนไขพิเศษในสนามต่อสู้:

- พื้นที่บางจุดจะมีกองลาวา หากยืนบนนั้นตอนเริ่มเทิร์น จะถูกเผาไหม้ต่อเนื่อง
- สามารถใช้สกิลธาตุน้ำหรือไอเทมประเภท "น้ำแข็ง" หรือ "ลม" เพื่อช่วยลด/ลบสนามลาวาได้บางส่วน

ของรางวัลเมื่อชนะ:

- **"ขนนกแห่งเปลวสุริยัน"** (Feather of Solar Flame) – ช่วยเพิ่มพลังโจมตีธาตุไฟ หรือปลดล็อกสกิลไฟสำหรับผู้เล่น

- ไอเทมพิเศษ เช่น เกราะกันไฟ อาวุธธาตุไฟ หรือวัตถุดิบตีบวก

---

2) "ราชินีหมาป่าน้ำแข็ง" (ธีมหิมะ/ดินแดนน้ำแข็ง)

ชื่อบอส:

- **"ราชินีหมาป่าน้ำแข็ง" (Frostfang Queen)**

ตำนาน/เนื้อเรื่องย่อ:

- ในดินแดนหิมะที่ไม่เคยละลาย "ราชินีหมาป่าน้ำแข็ง" ปกครองป่าเยือกแข็งด้วยเสียงคำรามทรงอำนาจ
- เธอถือครอง **"ขนนกแห่งหิมะนิรันดร์"** (Feather of Eternal Frost)
  ซึ่งมีพลังควบคุมความหนาวเหน็บและพายุหิมะ
- **ตำนานเล่าว่าเธอเคยเป็น "หมาป่าดวงจันทร์" จนกระทั่งได้รับพร (หรือคำสาป) จากเทพโบราณ ทำให้วิวัฒน์เป็นอสูรครึ่งวิญญาณที่ทรงภูมิปัญญา และเย็นชา**

รูปแบบ/คาแรกเตอร์:

- รูปร่างคล้ายหมาป่ายักษ์ขนสีขาวฟ้า มีเขาเล็ก ๆ และแผงคอเป็นเกล็ดน้ำแข็ง
- แววตาเยือกเย็น แต่ก็แฝงความดุร้ายและความฉลาด
- มีท่วงท่าราชินี สง่างาม น่าเกรงขาม

สกิลหลัก (ตัวอย่าง):

1. **Frost Howl** – หอนเสียงก้อง สร้างคลื่นกระแทกน้ำแข็ง โจมตีทั้งทีม และมีโอกาส "แช่แข็ง" ผู้เล่น ทำให้เทิร์นถัดไปเคลื่อนไหวช้าหรือทำได้จำกัด
2. **Snowstorm Veil** – สร้างพายุหิมะบดบังการมองเห็น ฝั่งผู้เล่นมีโอกาสพลาดการโจมตี (Accuracy ลดลง)
3. **Icicle Fang** – กระโดดจู่โจมเป้าหมายเดี่ยวด้วยเขี้ยวน้ำแข็ง สร้างความเสียหายสูงและอาจติดสถานะเลือดออก (Bleed) เพราะหนามน้ำแข็งเจาะผิวหนัง
4. **Glacial Domain (ท่าไม้ตาย)** – เปลี่ยนสนามต่อสู้เป็นทุ่งน้ำแข็งทั้งหมด (Field Effect) ส่งผลให้การเคลื่อนไหวของผู้เล่นช้าลง (Speed ลด) และสร้างดาเมจน้ำแข็งเป็นระยะ

เงื่อนไขพิเศษในสนามต่อสู้:

- หากผู้เล่นมีสกิลหรือไอเทมประเภท "ไฟ" หรือ "ลาวา" จะสามารถละลายน้ำแข็งในบางส่วน ลดผลกระทบของสนามน้ำแข็งได้
- เมื่อเลือดบอสลดต่ำกว่า 50% มีโอกาสเข้าสู่ "Frenzy Mode" เสริมพลังโจมตีและความเร็ว

ของรางวัลเมื่อชนะ:

- **"ขนนกแห่งหิมะนิรันดร์"** (Feather of Eternal Frost) – เพิ่มพลังเวทน้ำแข็ง หรือปลดล็อกสกิล/ดีบัฟจำพวกน้ำแข็ง
- อาจได้รับอาวุธน้ำแข็ง หรือไอเทมโจมตีธาตุเยือกแข็ง

---

3) "กงล้อจักรกลอมตะ" (ธีมเมืองเครื่องจักร/สตีมพังค์)

ชื่อบอส:

- **"กงล้อจักรกลอมตะ" (Clockwork Requiem)**
- หรือจะใช้ชื่อไทยตรง ๆ เช่น "ราชันแห่งฟันเฟือง"

ตำนาน/เนื้อเรื่องย่อ:

- ในเมืองหลวงแห่งจักรกลที่เต็มไปด้วยฟันเฟืองและไอน้ำ มีหอคอย "Clockspire" ตั้งตระหง่าน ที่ยอดหอคอยมีเครื่องจักรโบราณอันทรงพลังนามว่า **"กงล้อจักรกลอมตะ"**
- เป็นผู้เฝ้าขนนกที่เรียกว่า **"ขนนกแห่งกาลเวลา"** (Feather of Chronos) ซึ่งมีพลังหยุดหรือเร่งเวลาชั่วขณะ
- ผู้คนเชื่อว่าเครื่องจักรนี้เคยสร้างขึ้นเพื่อเป็นอาวุธปกป้องอาณาจักร แต่เมื่อเวลาผ่านไป มันกลับเห็นว่าเผ่าพันธุ์มนุษย์คือภัยคุกคาม เลยปิดกั้นตัวเองไว้พร้อมพัฒนาศักยภาพจนกลายเป็น "จักรกลอัจฉริยะ"

รูปแบบ/คาแรกเตอร์:

- ร่างกายโลหะขนาดใหญ่ มีแกนกลางเป็นกงล้อใหญ่หมุนตลอดเวลา รอบตัวมีฟันเฟืองเล็ก ๆ หมุนอยู่
- เคลื่อนไหวด้วยขา/ล้อจักรกล แต่สามารถเปลี่ยนรูปหรือยืดฟันเฟืองออกมาโจมตีระยะไกลได้
- มีแสงพลังงานสีฟ้า/เขียวบาง ๆ แสดงให้เห็นถึงแกนพลังเวทกล

สกิลหลัก (ตัวอย่าง):

1. **Time Distortion** – ทำให้เทิร์นโจมตีของบอสวนซ้ำ หรือข้ามเทิร์นผู้เล่นบางส่วน สุ่มเกิดผลเช่น บอสอาจได้เทิร์นเพิ่ม หรือผู้เล่นอาจเสีย 1 เทิร์น
2. **Gear Cannon** – แปลงฟันเฟืองเป็นปืนใหญ่พลังไอน้ำ ยิงกระสุนแรงสูงโดนเป้าหมายเดี่ยว หรือบางครั้งเป็น AoE กระสุนกระจาย
3. **Reconstruct** – ฟื้นฟูตัวเองโดยการดูดพลังงานจากฟันเฟืองโดยรอบ ทำให้ HP หรือเกราะบอสเพิ่มขึ้น
4. **Clockwork Catastrophe (ท่าไม้ตาย)** – เรียก "มหันตภัยฟันเฟือง" กลไกทั้งหมดในหอคอยจะทำงานพร้อมกัน สร้างความเสียหายหลายระลอก (ต่อเนื่อง 2-3 เทิร์น) และอาจสุ่มทำให้ผู้เล่นเกิดสถานะ "ติดขัด" (เคลื่อนไหวหรือใช้สกิลได้จำกัด)

เงื่อนไขพิเศษในสนามต่อสู้:

- พื้นที่บางส่วนมีสายพานหรือเฟืองหมุนที่อาจเปลี่ยนตำแหน่งผู้เล่นแบบสุ่มเมื่อจบเทิร์น
- หากผู้เล่นมีสกิล/อาวุธธาตุสายฟ้า (Lightning) สามารถทำให้ระบบบอส "ช็อต" ชะงักได้ในบางจังหวะ ลดอัตราใช้สกิลได้ชั่วคราว

ของรางวัลเมื่อชนะ:

- **"ขนนกแห่งกาลเวลา"** (Feather of Chronos) – เพิ่มความสามารถด้านความเร็ว/เทิร์น หรือปลดล็อกสกิลควบคุมเวลาชั่วคราว (เช่น เพิ่มเทิร์นตัวเองหรือลดเทิร์นศัตรู)
- อาจได้รับอุปกรณ์จักรกลหายาก เช่น แขนกลเพิ่มพลังโจมตี หรือตัวเร่งฟื้น MP

มอนเตอร์ในแต่ละธีม

1) ธีมภูเขาไฟ/ดินแดนเพลิง (Fire/Volcano Theme)

1. **Flame Imp**
    - **คอนเซ็ปต์:** ซาตานตัวจิ๋วผิวแดง มีเขาสั้นๆ และถือคบเพลิงเล็ก ๆ วิ่งพล่านไปทั่วบริเวณ
    - **สกิลหลัก:**
        - *Fire Toss:* ขว้างลูกไฟระยะไกล สร้างความเสียหายธาตุไฟเบื้องต้น
        - *Torch Poke:* โจมตีระยะประชิดด้วยคบเพลิง ทำให้ติดสถานะ "เผาไหม้ (Burn)" อ่อน ๆ
2. **Lava Slime**
    - **คอนเซ็ปต์:** สไลม์ก้อนเดือดที่ภายในเป็นลาวา เหนียวหนืดและร้อนระอุ
    - **สกิลหลัก:**
        - *Molten Splash:* สะบัดเนื้อลาวาใส่เป้าหมาย มีโอกาสทำให้ติดสถานะ "ร้อนจัด" (ลดพลังป้องกันช่วงสั้น ๆ)
        - *Split:* เมื่อ HP ลดถึงครึ่ง อาจแบ่งร่างเป็นสไลม์ลาวาขนาดเล็กสองตัว (HP/พลังโจมตีลดลง แต่จำนวนเพิ่มขึ้น)
3. **Magma Wolf**
    - **คอนเซ็ปต์:** หมาป่าผิวหนังแตกร้าว มีลาวาเรืองแสงไหลซึมตามรอยแตก บางครั้งไฟลุกบริเวณอุ้งเท้าและดวงตา
    - **สกิลหลัก:**
        - *Inferno Fang:* กระโดดกัดอย่างรุนแรง เสริมด้วยพลังไฟ เผาไหม้ศัตรูเมื่อโจมตีสำเร็จ
        - *Howl of Embers:* หอนกึกก้อง สร้างดีบัฟลดพลังป้องกันให้ศัตรูรอบข้าง และเพิ่มพลังโจมตีให้ตนเองและพวกพันธมิตรประเภท "หมาป่าไฟ"
4. **Igneous Golem**
    - **คอนเซ็ปต์:** โกเลมหินผสมลาวา รูปร่างใหญ่เทอะทะ พลังป้องกันสูง
    - **สกิลหลัก:**
        - *Rocky Pound:* ทุบพื้นอย่างแรง ทำดาเมจ AoE ระยะใกล้

- *Molten Core:* สะสมพลังที่แกนกลาง
  เมื่อถูกโจมตีด้วยธาตุไฟจะยิ่งร้อนและปลดปล่อยคลื่นความร้อนสวนกลับ (เล็กน้อย)

---

2) ธีมหิมะ/ดินแดนน้ำแข็ง (Snow/Ice Theme)

1. **Snow Goblin**
   - **คอนเซ็ปต์:** ก็อบลินตัวเล็ก ๆ ในชุดขนสัตว์ หูยาว ถือหอกน้ำแข็งหรือกระบองกระดูกที่ปกคลุมด้วยน้ำค้างแข็ง
   - **สกิลหลัก:**
     - *Ice Javelin:* ขว้างหอกน้ำแข็งใส่เป้าหมายระยะไกล มีโอกาสลดความเร็ว (Speed) เป้าหมาย
     - *Bone Chill:* เคาะกระบองทำให้เศษน้ำแข็งกระจาย ตัวถูกโจมตีอาจติดสถานะ “ชา” (ลดพลังโจมตีชั่วคราว)
2. **Frost Wisp**
   - **คอนเซ็ปต์:** ลูกไฟเยือกแข็งล่องลอยมีใบหน้าเล็ก ๆ หรือรูปร่างคล้ายวิญญาณ แผ่ไอเย็นเป็นประกายสีฟ้าขาว
   - **สกิลหลัก:**
     - *Chilling Touch:* พุ่งเข้าหาเป้าหมาย
       สร้างความเสียหายน้ำแข็งปานกลางและลดความเร็วของศัตรู
     - *Snowfall Aura:* แผ่รัศมีไอเย็นรอบตัว รักษา HP ของ Wisp (เล็กน้อย) และลด HP ของศัตรูในระยะอย่างต่อเนื่อง (หิมะกัด)
3. **Ice Drake**
   - **คอนเซ็ปต์:** มังกรย่อส่วน (ลักษณะกึ่งมังกรกึ่งกิ้งก่า) ปีกเป็นแผ่นน้ำแข็งบาง ๆ หางเรียวยาว
   - **สกิลหลัก:**
     - *Freezing Breath:* พ่นลมหายใจเย็นจัดเป็นกรวยด้านหน้า สร้างดาเมจและโอกาสติดสถานะ “แช่แข็ง” ชั่วครู่
     - *Glide Slash:* ร่อนตัวพุ่งเฉือนเป้าหมายด้วยกรงเล็บน้ำแข็ง
4. **Polar Yeti**
   - **คอนเซ็ปต์:** เยติร่างยักษ์ที่อยู่ในถ้ำหิมะ ขนปุกปุยสีขาว ปลายขนเปื้อนน้ำแข็งเกรอะกรัง
   - **สกิลหลัก:**
     - *Frost Pound:* ทุบเป้าหมายด้วยแขนใหญ่ สร้างดาเมจสูงระยะประชิด
     - *Snow Boulder:* โยนก้อนหิมะยักษ์ สร้างความเสียหาย AoE และมีโอกาส “สตัน” (ทำให้ศัตรูเสียเทิร์น)

3) ธีมเมืองเครื่องจักร/สตีมพังค์ (Steampunk/Clockwork Theme)

1. **Rusty Automaton**
    - **คอนเซ็ปต์:** หุ่นจักรกลทรงมนุษย์ขนาดเล็ก พังโทรุดโทรม มีเสียงเฟืองดังเอี๊ยดอ๊าด
    - **สกิลหลัก:**
        - *Metal Punch:* โจมตีระยะประชิดด้วยหมัดโลหะ
        - *Overheat Spark:* เมื่อ HP ใกล้หมด มีโอกาสปล่อยประกายไฟช็อตศัตรูที่อยู่ใกล้
2. **Steam Spider**
    - **คอนเซ็ปต์:** แมงมุมจักรกลขาเหล็กหลายข้าง มีปล่องปล่อยไอน้ำเหนือลำตัว
    - **สกิลหลัก:**
        - *Steam Jet:* พ่นไอน้ำร้อนสร้างความเสียหายปานกลางและอาจทำให้ศัตรูมึน (Accuracy ลดลง)
        - *Gear Web:* ปล่อยใยฟันเฟืองยึดศัตรู ทำให้เคลื่อนไหวช้าหรือไม่สามารถหลบได้ชั่วคราว
3. **Mechanical Hound**
    - **คอนเซ็ปต์:** สุนัขโลหะสี่ขา มีตาเป็นเลนส์สีแดง เร็วและดุดัน
    - **สกิลหลัก:**
        - *Gear Bite:* กัดรุนแรง สร้างความเสียหายกายภาพสูง
        - *High-Pitch Whirr:* ปล่อยเสียงเครื่องจักรความถี่สูง รบกวนศัตรู ลดค่าป้องกันหรือความแม่นยำ
4. **Battery Mantis**
    - **คอนเซ็ปต์:** ตั๊กแตนตำข้าวจักรกล ถือแบตเตอรี่ง่ามแหลมเป็นแขนเคียว 2 ข้าง ปีกโลหะรูปใบพัด
    - **สกิลหลัก:**
        - *Charged Slash:* โจมตีด้วยแขนเคียวพร้อมปล่อยกระแสไฟฟ้า หากโดนจะติดสถานะ “ช็อต” (ตัด MP หรือทำให้เทิร์นถัดไปช้าลง)
        - *Power Surge:* ดูดพลังงานจากแบตเตอรี่ เสริมพลังโจมตีและความเร็วของตัวเองชั่วครู่

Class(ยังไม่สรุป)

-“Damage Over Time” (DoT Effects)

Class BaseDotEffect

FireBurn // ลด Hp เรื่อยๆ หรือลด DEF เป้าหมายเล็กน้อยเมื่อถูกไฟไหม้

Poison // ลด HP สูงขึ้นตามเวลาที่ติด

Bleeding // เน้นลด HP อย่างสม่ำเสมอ และลด DEF หรือ SPD

```
class BaseDotEffect {
    // คุณสมบัติพื้นฐาน
    float damagePerTurn;
    int duration; // ระยะเวลา (เทิร์น/วินาที)
    string effectName;
    // เมธอดหลัก
    void ApplyEffect(Character target);
    void RemoveEffect(Character target);
    void TickEffect(Character target); // เรียกในแต่ละเทิร์น
}
```

-“Debuff”

class BaseDebuff

StunDebuff // ทำให้เป้าหมายข้ามเทิร์น / ไม่สามารถเคลื่อนไหวได้

FreezeDebuff // ลด SPD อย่างมาก

SlowDebuff // ลด SPD

BlindDebuff // ลด Accuracy ทำให้โจมตีพลาดง่ายขึ้น

```
class BaseDebuff {
    string debuffName;
    int duration;
    // คุณสมบัติอื่น ๆ เช่น ลด ATK, DEF, SPD, etc.
```

```
    void ApplyDebuff(Character target);
    void RemoveDebuff(Character target);
    void OnTurnStart(Character target);
    void OnTurnEnd(Character target);
}
```

-“Buff”

class BaseBuff

AttackUp

DefenseUp

RegenBuff (ฟื้น HP เล็กน้อยทุกเทิร์น)

```
class BaseBuff {
    string buffName;
    int duration;
    // คุณสมบัติ เช่น เพิ่ม ATK, DEF, SPD, etc.
    void ApplyBuff(Character target);
    void RemoveBuff(Character target);
    void OnTurnStart(Character target);
    void OnTurnEnd(Character target);
}
```

-“FieldEffect”

```
class BaseFieldEffect {
    string effectName;      // ชื่อเอฟเฟค เช่น "Lava Field", "Blizzard Storm"
    int duration;           // ระยะเวลา (หน่วยเป็นเทิร์นหรือวินาที)
    bool isPermanent;       // บางเอฟเฟคอาจอยู่ถาวรจนกว่าจะถูกลบด้วยสกิล/  เงื่อนไขพิเศษ

    // เมธอดหลัก
    void ApplyEffectToField();
    void RemoveEffectFromField();
    void OnTurnStart();
    void OnTurnEnd();
}
```

-ApplyEffectToField(): เรียกตอนเริ่มใช้สกิล หรือเมื่อติดตั้งเอฟเฟคลงสนาม

-RemoveEffectFromField(): เรียกเมื่อเอฟเฟคหมดเวลา หรือมีเงื่อนไขยกเลิก

-OnTurnStart() / OnTurnEnd(): ถ้าเอฟเฟคมีการอัปเดตต่อเทิร์น เช่น ทุกเทิร์นลด HP ทั้งหมด 5 หน่วย เป็นต้น

ตัวอย่างเอฟเฟคย่อย

- **LavaFieldEffect**: เปลี่ยนฉากต่อสู้เป็นพื้นลาวา
  ทุกเทิร์นสร้างความเสียหายไฟให้ผู้เล่นและศัตรู (ยกเว้นบางตัวที่ทนไฟ)
- **BlizzardFieldEffect**: ลด SPD ของทุกฝ่ายลง,
  ทุกเทิร์นมีโอกาสทำให้ตัวละครติดสถานะแช่แข็ง
- **TimeDistortionEffect**: บอสอาจได้รับเทิร์นพิเศษ หรือผู้เล่นมีโอกาสข้ามเทิร์น เป็นต้น

-Player

```
class Player {
        // ข้อมูลพื้นฐานของผู้เล่น
        int level;          // เลเวลปัจจุบัน
        int currentXP;      // ประสบการณ์ที่สะสมได้ในเลเวลปัจจุบัน
        int xpToNextLevel;  // คะแนนประสบการณ์ที่ต้องการสำหรับเลเวลถัดไป

        // สเตตัสพื้นฐานของผู้เล่น (ค่า Base)
        int baseHP;
        int baseAtk;
        int baseDef;
        int baseSpd;

        // สเตตัสปัจจุบัน (อาจคำนวณจาก base + modifier จากเลเวล)
        int hp;
        int atk;
        int def;
        int spd;

        // Constructor: กำหนดค่าเริ่มต้น
        Player() {
            level = 1;
            currentXP = 0;
            xpToNextLevel = computeXPThreshold(level);  // กำหนด xp ที่ต้องการในเลเวลแรก

            // กำหนดค่า base เริ่มต้น (ตัวอย่าง)
            baseHP = 100;
            baseAtk = 20;
```

```
    baseDef = 15;
    baseSpd = 10;

    // คำนวณสเตตัสเริ่มต้น
    recalcStats();
}

// ฟังก์ชันเพิ่ม XP ให้กับผู้เล่น
void addXP(int xp) {
    currentXP += xp;
    // ตรวจสอบว่าเพียงพอสำหรับการเลเวลขึ้นหรือไม่
    while (currentXP >= xpToNextLevel) {
        levelUp();
    }
}

// ฟังก์ชันเลเวลขึ้น
void levelUp() {
    level++;
    currentXP -= xpToNextLevel;          // หัก XP ที่ใช้ในการเลเวลขึ้น
    xpToNextLevel = computeXPThreshold(level); // คำนวณ XP ที่ต้องการสำหรับเลเวลถัดไป
    recalcStats();                       // คำนวณสเตตัสใหม่ตามเลเวล
    // สามารถเพิ่มข้อความแจ้งเตือนหรือเอฟเฟคเลเวลขึ้นได้ที่นี่
    print("เลเวลขึ้น! ตอนนี้เลเวล: " + level);
}

// ฟังก์ชันคำนวณ XP ที่ต้องการสำหรับเลเวลถัดไป (สามารถปรับสูตรได้)
int computeXPThreshold(int currentLevel) {
    // ตัวอย่าง: xpToNextLevel = 100 * currentLevel^1.2 (หรือสูตรอื่นตามความเหมาะสม)
    return int(100 * pow(currentLevel, 1.2));
}

// ฟังก์ชันคำนวณสเตตัสของผู้เล่นใหม่เมื่อเลเวลขึ้น
void recalcStats() {
    // ตัวอย่าง: เพิ่มสเตตัส 5% ต่อเลเวล (หรือปรับสูตรได้ตามต้องการ)
    hp = baseHP + (baseHP * (level - 1) * 0.05);
    atk = baseAtk + (baseAtk * (level - 1) * 0.05);
```

```
            def = baseDef + (baseDef * (level - 1) * 0.05);
            spd = baseSpd + (baseSpd * (level - 1) * 0.05);
        }
    }
```

-คลาส DifficultyManager

```
    // กำหนด Enum สำหรับระดับความยาก
    enum DifficultyLevel {
       EASY,
       NORMAL,
       HARD,
       NIGHTMARE
    }

    class DifficultyManager {
            DifficultyLevel difficulty;

            // ตัวคูณสำหรับสเตตัสต่างๆ ของมอนสเตอร์
            float hpMultiplier;
            float atkMultiplier;
            float defMultiplier;
            float spdMultiplier;
        // ตัวคูณสำหรับ Exp ที่ได้รับ
        float expMultiplier;

            // Constructor: รับระดับความยากที่เลือกเข้ามา
            DifficultyManager(DifficultyLevel selectedDifficulty) {
                difficulty = selectedDifficulty;
                setMultipliers();
                setExpMultiplier();
            }


        // ฟังก์ชันตั้งค่าตัวคูณสำหรับ Exp
        void setExpMultiplier() {
            if (difficulty == DifficultyLevel.EASY) {
                expMultiplier = 1.0;
            } else if (difficulty == DifficultyLevel.NORMAL) {
```

```
        expMultiplier = 1.2;
    } else if (difficulty == DifficultyLevel.HARD) {
        expMultiplier = 1.5;
    } else if (difficulty == DifficultyLevel.NIGHTMARE) {
        expMultiplier = 2.0;
    }
}
// ฟังก์ชันตั้งค่าตัวคูณตามระดับความยาก
void setMultipliers() {
    if (difficulty == DifficultyLevel.EASY) {
        hpMultiplier = 1.0;
        atkMultiplier = 1.0;
        defMultiplier = 1.0;
        spdMultiplier = 1.0;
    } else if (difficulty == DifficultyLevel.NORMAL) {
        hpMultiplier = 1.2;
        atkMultiplier = 1.2;
        defMultiplier = 1.2;
        spdMultiplier = 1.1;
    } else if (difficulty == DifficultyLevel.HARD) {
        hpMultiplier = 1.5;
        atkMultiplier = 1.5;
        defMultiplier = 1.5;
        spdMultiplier = 1.3;
    } else if (difficulty == DifficultyLevel.NIGHTMARE) {
        hpMultiplier = 2.0;
        atkMultiplier = 2.0;
        defMultiplier = 2.0;
        spdMultiplier = 1.5;
    }
}

// ฟังก์ชันสำหรับปรับสเตตัสของมอนสเตอร์ที่สร้างใหม่
// รับค่า base stats ของมอนสเตอร์แล้วคืนค่า stats ที่ปรับตามตัวคูณ
Monster scaleMonsterStats(Monster baseMonster) {
    Monster scaledMonster = new Monster();
    scaledMonster.hp  = baseMonster.hp  * hpMultiplier;
```

```
            scaledMonster.atk = baseMonster.atk * atkMultiplier;
            scaledMonster.def = baseMonster.def * defMultiplier;
            scaledMonster.spd = baseMonster.spd * spdMultiplier;
            // ปรับสเตตัสอื่นๆ ได้ตามต้องการ
            return scaledMonster;
        }
    }
```

-คลาสพื้นฐานสำหรับ Enemy

```
class BaseEnemy {
    string name;
    int hp;
    int atk;
    int def;
    int spd;
    List<Skill> skills;      // สกิลที่ตัวละครใช้ได้

    // เมธอดพื้นฐาน
    void takeDamage(int damage) { ... }
    void useSkill(Skill skill, BaseEnemy target) { ... }
    void onTurnStart() { ... }
    void onTurnEnd() { ... }
}
```

-คลาสสำหรับบอส (Boss Classes)

```
class BaseBoss extends BaseEnemy {
    FieldEffect fieldEffect; // เอฟเฟคที่กระทบทั้งสนาม

    // เมธอดที่อาจเรียกใช้ FieldEffect เมื่อใช้สกิลพิเศษ
    void activateFieldEffect() {
        fieldEffect.applyEffectToField();
    }
```

```
}
```

-บอสแต่ละธีม

- บอสธีมภูเขาไฟ/ดินแดนเพลิง — “อัคคีวิญญาณ”

```
class VolcanoBoss extends BaseBoss {
  VolcanoBoss() {
    name = "อัคคีวิญญาณ";
    hp = 1000;
    atk = 150;
    def = 100;
    spd = 80;
    skills = [ EruptionSmash, MoltenShield, FlameWave, LavaSurge ];
    fieldEffect = new LavaFieldEffect(duration: 3, damagePerTurn: 20);
  }
}
```

- บอสธีมหิมะ/ดินแดนน้ำแข็ง — “ราชินีหมาป่าน้ำแข็ง”

```
class IceBoss extends BaseBoss {
  IceBoss() {
    name = "ราชินีหมาป่าน้ำแข็ง";
    hp = 900;
    atk = 130;
    def = 110;
    spd = 70;
    skills = [ FrostHowl, SnowstormVeil, IcicleFang, GlacialDomain ];
    fieldEffect = new BlizzardFieldEffect(duration: 3, spdReduction: 10);
  }
}
```

- บอสธีมสตีมพังค์ — “กงล้อจักรกลอมตะ”

```
class SteampunkBoss extends BaseBoss {
        SteampunkBoss() {
          name = "กงล้อจักรกลอมตะ";
          hp = 1100;
          atk = 140;
          def = 120;
          spd = 90;
          skills = [ TimeDistortion, GearCannon, Reconstruct, ClockworkCatastrophe ];
          fieldEffect = new TimeDistortionEffect(duration: 2, effectDescription: "เพิ่มเทิร์นให้บอส");
```

```
        }
}
```

-คลาสสำหรับมอนสเตอร์ (Monster Classes)

1) มอนสเตอร์ธีมภูเขาไฟ/ดินแดนเพลิง (4 ตัว)

Flame Imp

```
class FlameImp extends BaseEnemy {
        FlameImp() {
                name = "Flame Imp";
                hp = 100;
                atk = 30;
                def = 10;
                spd = 50;
                skills = [ FireToss, TorchPoke ];
        }
}
```

Lava Slime

```
class LavaSlime extends BaseEnemy {
        LavaSlime() {
                name = "Lava Slime";
                hp = 150;
                atk = 20;
                def = 15;
                spd = 30;
                skills = [ MoltenSplash, Split ];
        }
}
```

Magma Wolf

```
class MagmaWolf extends BaseEnemy {
        MagmaWolf() {
                name = "Magma Wolf";
                hp = 180;
                atk = 40;
```

```
                def = 20;
                spd = 60;
                skills = [ InfernoFang, HowlOfEmbers ];
        }
}
```

Igneous Golem

```
class IgneousGolem extends BaseEnemy {
        IgneousGolem() {
                name = "Igneous Golem";
                hp = 250;
                atk = 35;
                def = 40;
                spd = 20;
                skills = [ RockyPound, MoltenCore ];
        }
}
```

2) มอนสเตอร์ธีมหิมะ/ดินแดนน้ำแข็ง (4 ตัว)

Snow Goblin

```
class SnowGoblin extends BaseEnemy {
        SnowGoblin() {
                name = "Snow Goblin";
                hp = 90;
                atk = 25;
                def = 12;
                spd = 55;
```

```
                skills = [ IceJavelin, BoneChill ];
        }
}
```

Frost Wisp

```
class FrostWisp extends BaseEnemy {
        FrostWisp() {
                name = "Frost Wisp";
                hp = 80;
                atk = 20;
                def = 10;
                spd = 65;
                skills = [ ChillingTouch, SnowfallAura ];
        }
}
```

Ice Drake

```
class IceDrake extends BaseEnemy {
        IceDrake() {
                name = "Ice Drake";
                hp = 200;
                atk = 45;
                def = 25;
                spd = 50;
                skills = [ FreezingBreath, GlideSlash ];
        }
}
```

Polar Yeti

```
class PolarYeti extends BaseEnemy {
        PolarYeti() {
                name = "Polar Yeti";
                hp = 220;
                atk = 50;
                def = 30;
                spd = 35;
                skills = [ FrostPound, SnowBoulder ];
        }
}
```

3) มอนสเตอร์ธีมสตีมพังค์ (4 ตัว)

Rusty Automaton

```
class RustyAutomaton extends BaseEnemy {
        RustyAutomaton() {
                name = "Rusty Automaton";
                hp = 120;
                atk = 35;
                def = 20;
                spd = 40;
                skills = [ MetalPunch, OverheatSpark ];
        }
}
```

Steam Spider

```
class SteamSpider extends BaseEnemy {
        SteamSpider() {
                name = "Steam Spider";
                hp = 110;
                atk = 30;
                def = 18;
                spd = 45;
                skills = [ SteamJet, GearWeb ];
        }
}
```

Mechanical Hound

```
class MechanicalHound extends BaseEnemy {
        MechanicalHound() {
                name = "Mechanical Hound";
                hp = 130;
                atk = 40;
                def = 22;
                spd = 50;
                skills = [ GearBite, HighPitchWhirr ];
        }
}
```

Battery Mantis

```
class BatteryMantis extends BaseEnemy {
        BatteryMantis() {
                name = "Battery Mantis";
                hp = 100;
                atk = 38;
                def = 15;
                spd = 55;
                skills = [ ChargedSlash, PowerSurge ];
        }
}
```

-Base Skill Class

```
class Skill {
```

```
    string name;          // ชื่อสกิล
    int cost;             // ค่าใช้จ่าย (เช่น MP หรือค่าใช้จ่ายอื่น ๆ)
    int cooldown;         // คูลดาวน์ที่กำหนดไว้ (จำนวนเทิร์น)
    int currentCooldown;  // ตัวนับคูลดาวน์ที่เหลือ
    string description;   // รายละเอียดสกิล

    // Constructor
    Skill(string name, int cost, int cooldown, string description) {
        this.name = name;
        this.cost = cost;
        this.cooldown = cooldown;
        this.currentCooldown = 0;
        this.description = description;
    }

    // ใช้สกิล (ต้อง override ใน subclass ที่ระบุการทำงานเฉพาะ)
    virtual void useSkill(Character user, Character target) {
        // ตรวจสอบว่า cooldown เป็น 0 หรือไม่
        if(currentCooldown == 0 && user.hasEnoughCost(cost)) {
            // ดำเนินการใช้สกิล
            execute(user, target);
            currentCooldown = cooldown; // รีเซ็ตคูลดาวน์
        }
    }

    // ฟังก์ชันเฉพาะสำหรับสกิล (ให้ override ใน subclass)
    virtual void execute(Character user, Character target) { }

    // ลดคูลดาวน์ในแต่ละเทิร์น
    void reduceCooldown() {
        if(currentCooldown > 0) {
            currentCooldown--;
        }
    }
}
```

ตัวอย่าง Subclass: Active Skill (เช่น FireToss) คิดว่าจริงๆต้อง - กับ Def ด้วย

```
class FireToss extends Skill {
```

```
    int baseDamage;

    FireToss() : super("Fire Toss", cost: 10, cooldown: 2, "ขว้างไฟทำความเสียหายระยะไกล พร้อมโอกาสติด
Burn") {
        baseDamage = 30;
    }

    // Override execute เพื่อกำหนดการทำงานของสกิล
    override void execute(Character user, Character target) {
        // คำนวณความเสียหาย (อาจคำนวณจาก ATK ของ user)
        int damage = baseDamage + user.atk * 0.5;
        target.takeDamage(damage);
        // เพิ่มโอกาสติด Burn (สามารถใช้ EffectManager เพิ่ม DoT)
        target.applyEffect(new FireBurnEffect(damagePerTurn: 5, duration: 3));
    }
}
```

-UI Screens

-UI_TitleScreen

-UI_DifficultySelection

- หน้าจอเลือกความยาก

-UI_CombatScreen (HUD)

- แสดง HP/MP/Lv/EXP ของ Player และ Enemy ในระหว่างต่อสู้
- ปุ่ม “ต่อสู้ / ใช้ไอเท็ม / หนี” หรือแสดงรายการสกิล
- อาจตั้งชื่อ UI_BattleHUD, UI_CombatScreen, หรือ Scene_Battle

-UI_BattleMenu

- ไว้จัดการเลือกสกิล, โจมตี, บัฟ, ไอเท็ม ฯลฯ

-UI_BattleResult

- หลังสู้จบหรือปราบบอสเสร็จ จะแสดงรางวัลหรือสรุปคะแนน

-UI_Town

- หน้าจอเมือง (Hub) ที่มีการพักผ่อน, ร้านค้า, รับเควสต์, อัปเกรดต่าง ๆ

-UI_Inventory

- แยกต่างหากจาก TownScreen ก็ทำเป็นหน้าต่าง Popup สำหรับดูไอเท็ม, อัปสเตตัส

-EffectManager

- รับหน้าที่จัดการเพิ่ม/ลบ Effect (DoT/Buff/Debuff) ให้กับตัวละคร และเรียก Tick ในแต่ละเทิร์น

-UIManager

- ดูแลการสลับหน้าจอ (MainMenu → Difficulty → InGameHUD → Town → ฯลฯ) โดยใช้ State Machine หรือ Scene Controller

-GameManager

- ดูแลโฟลว์หลักของเกม เช่น เมื่อจบด่านแล้วไปหน้า Reward, จาก Reward ไป Town, จาก Town เลือกด่านถัดไป เป็นต้น