

รายงานความก้าวหน้าวิชา CE Project

ครั้งที่ 2

ระหว่างวันที่ 28 ส.ค. 65 ถึงวันที่ 09 ก.ย. 65

1. ชื่อโครงการ (อังกฤษ) The Rouge-Like Action RPG Game Development
2. การดำเนินงานมีความก้าวหน้า 22% (ใช้ค่า % **Complete** จาก MS Project)

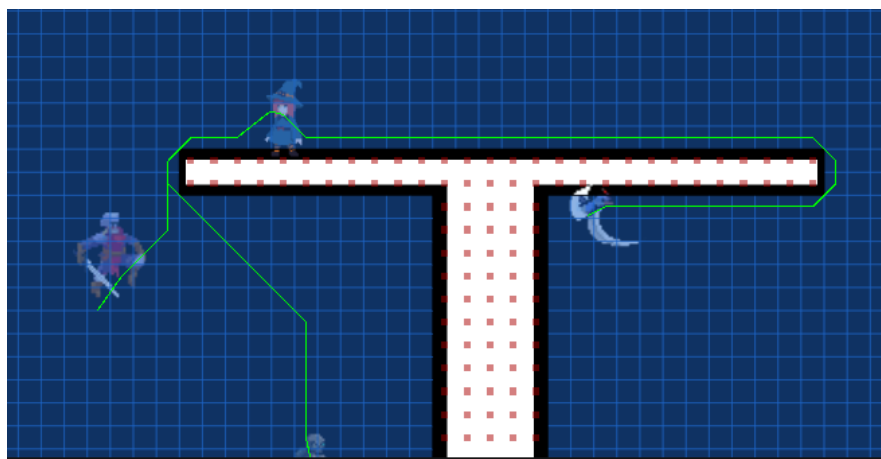
มีความก้าวหน้าเพิ่มขึ้นจากรายงานความก้าวหน้า ครั้งก่อน 22%

☒ เร็วกว่าแผน 1 วัน ☐ ช้ากว่าแผน วัน

3. รายละเอียดความก้าวหน้า

ศัตรู (Enemy)

- ทำ Template ของศัตรูและ Ai ของศัตรู : ส่วนนี้เราสามารถพัฒนาได้อย่างราบรื่นแต่ยังไม่เสร็จสมบูรณ์นักโดยในปัจจุบันระบบ Ai ของศัตรูสามารถพัฒนาไปได้ถึงการหาเส้นทางที่จะวิ่งไปหาผู้เล่นได้แล้ว แต่ปัจจุบันยังเป็นการเคลื่อนที่ไปหาผู้เล่นเพียงอย่างเดียว (ยังไม่สามารถโจมตี , ออก Action ต่างๆที่ไม่ซ้ำซาก) ซึ่งในส่วนนี้จะพัฒนาต่อไป



ภาพแสดงเส้นทางต่างๆที่ศัตรูจะใช้เดินทางไปยังผู้เล่น

- การทำ Animation ต่างๆของศัตรู : ในส่วนนี้พบปัญหาเล็กน้อยแต่ได้แก้ไขไปเรียบร้อยแล้ว สามารถดูและ โยง Animation ต่างๆนำไปแสดงผลในเกมได้แล้ว

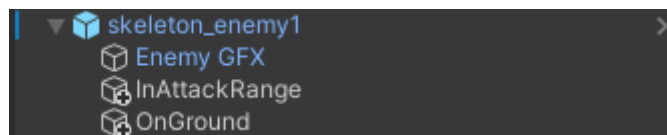
จัดทำเอกสารการออกแบบเกม (Game Design Document)

จัดทำเอกสาร การออกแบบเกม (Game Design Document) ซึ่งจะเป็นเอกสารที่บอกรายละเอียดต่างๆของเกม ว่าเกมที่จะพัฒนามี Concept อย่างไร และรายละเอียดการออกแบบปพลิเคชันต่างๆ เช่น ระบบของเกม เนื้อเรื่อง ตัวละคร เป้าหมายของเกม ซึ่งตัวเอกสารที่จัดทำสามารถเข้าไปดูได้ที่ [Github](#) ของทางผู้พัฒนา

4. ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

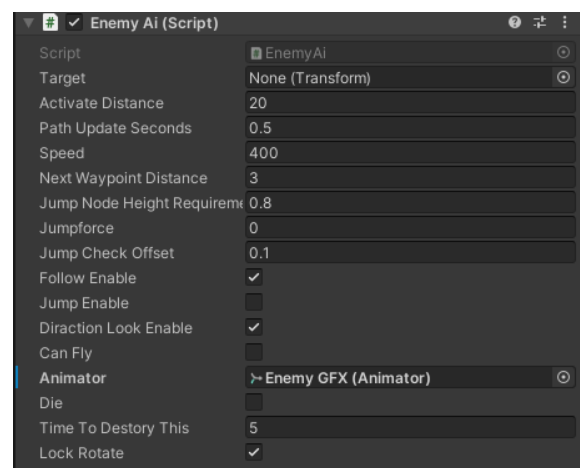
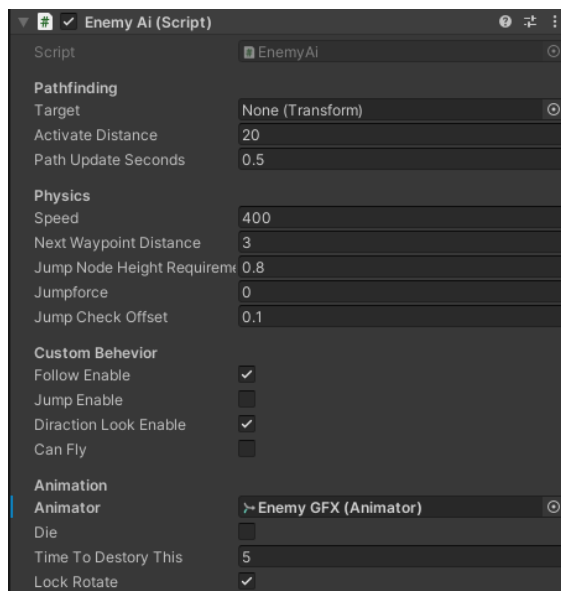
ศัตรู (Enemy)

ปัญหาที่พบ : ทำ Template ของศัตรูจากที่ออกแบบไว้ตอนแรกคิดว่าจะใช้วิธีแบ่งเป็น Object แต่ละประเภท แต่หลังจากทำตามที่ทำออกแบบจนเสร็จแล้วพบว่า Object ของแต่ละศัตรูแต่ละตัวจะประกอบไปด้วย Object ย่อยและ Object ย่อยก็จะมี Script ย่อยลงไปอีก



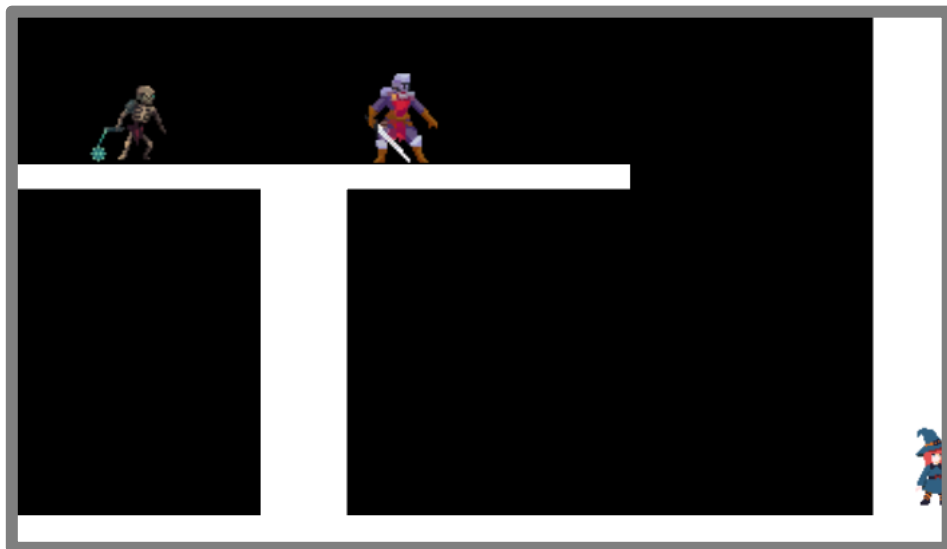
ซึ่งผู้พัฒนาได้ออกความเห็นว่าการ Script ต่างๆถูกแบบย่อยออกจากกันเกินความจำเป็นจะทำให้ความยุ่งยากในช่วง Parameter tuning (การปรับบาลานซ์ของเกม) ในภายหลัง

วิธีแก้ปัญหา : ใช้วิธีรวม Parameter ต่างๆไว้ใน Script เดียวโดยนำ Tag [Header] เข้ามาช่วยวิธีนี้จะทำให้สามารถแก้ไข Parameter ต่างๆได้โดยง่ายและจะสามารถแยกประเภทของ Parameter ได้อีกด้วย



ภาพเปรียบเทียบระหว่างการใช้ Tag Header และไม่ใช่

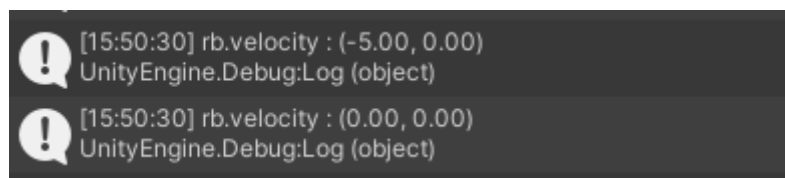
ปัญหาที่พบ : บางครั้งตัวละครจะเคลื่อนที่ทะลุฉากหรือสิ่งที่ไม่ควรทะลุไปได้เช่นกำแพง



รูปที่ตัวละครทะลุไปยังหลังกำแพง

วิธีแก้ปัญหา : ในช่วงที่เกิดปัญหาเราสามารถใช้วิธีการเคลื่อนที่แบบ Transform.Translate วิธีนี้จะเป็นการเคลื่อนที่แบบไม่สนใจฟิสิกส์ของ Unity คล้ายๆกับการบังคับให้ Object นั้นไปยังพิกัดนั้นๆ (อ้างอิง [Link](#))

หลังจากที่เราศึกษาวิธีการเคลื่อนที่ด้วยเทคนิคอื่นแล้วพบว่ามียุทธวิธีอีก 2 อย่างคือ rb.velocity และ rb.addforce โดยครั้งนี้เราเลือก rb.velocity เนื่องจากเทคนิคนี้เป็นการใส่แรงและทิศทางเมื่อเรากดปุ่มและจะปรับให้แรงนั้นเป็น 0 เมื่อเราปล่อยปุ่ม



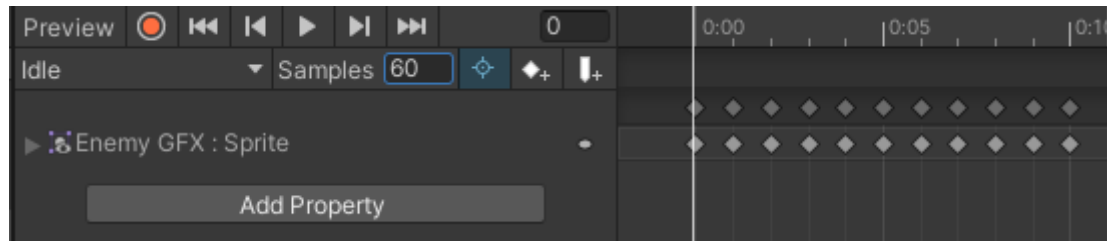
ภาพที่แสดงค่า velocity หลังจากปล่อยปุ่ม

โดย rb.addforce จะเป็นการใส่แรงและทิศทางไปเหมือนกันแต่ถ้าหากปล่อยปุ่มแรงนั้นจะไม่เหลือ 0 แต่จะเป็นการค่อยๆผ่อนแรงนั้นๆลงแทน

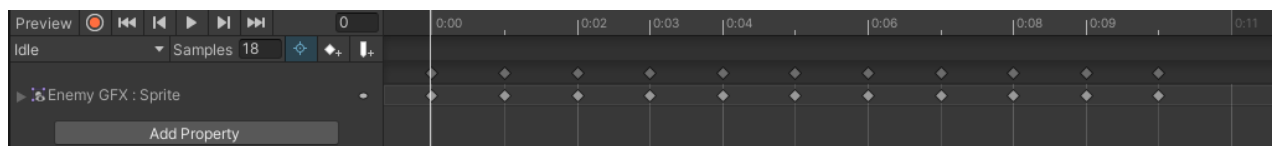
Animation

ปัญหาที่พบ : Frame ของตัวละครต่างๆมีค่อนข้างน้อยและเมื่อนำมาทำ Animation จะทำให้ Animation นั้นๆ เร็วเกินไปจนดูไม่เป็นธรรมชาติ

วิธีแก้ปัญหา : นำระบบ Sample rate ของ Animation เข้ามาช่วยโดยระบบนี้จะช่วยหน่วงความเร็วของ Animation ตามความเร็วที่เรากำหนดจึงทำให้ Animation render ได้อย่างสบายตา



ภาพเมื่อ Samples rate เป็น 60 (ค่าเริ่มต้น)



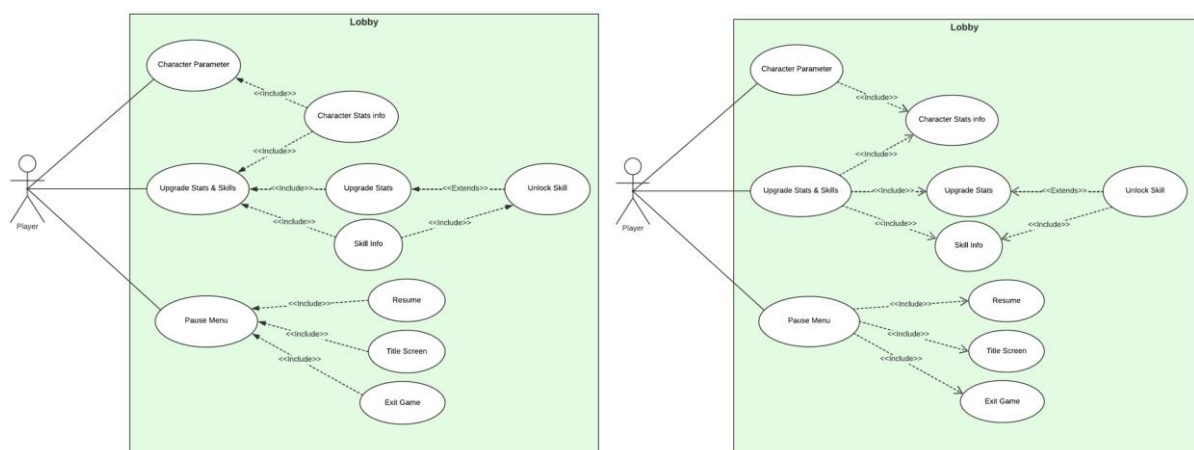
ภาพเมื่อ Samples rate เป็น 18 (ค่าที่ปรับแล้วคิดว่าลงตัว)

ทั้ง 2 ภาพจะแสดงการเล่นของ Animation ใน Time scale เท่าๆ กัน

เอกสารการออกแบบเกม (Game Design Document)

ปัญหาที่พบ : ระหว่างทำเอกสารได้มีการออกแบบ Use Case Diagram ของหน้าจอ User Interface หลังจากได้ส่งให้อาจารย์ที่ปรึกษาตรวจพบว่า มีการใช้ตัว Relationship ระหว่าง Use Case ได้ไม่ถูกต้องตามหลักการเขียน

วิธีแก้ปัญหา : อาจารย์ที่ปรึกษาได้ให้คำแนะนำในการเขียน Use Case Diagram และแนะนำให้ศึกษาและเปรียบเทียบกับตัวมาตรฐานการเขียน Diagram ต่างๆ (อ้างอิงเว็บเปรียบเทียบ [Link](#))



ภาพ Use case diagram ที่ใช้ Relationship ผิด(ซ้าย) และ Use case diagram หลังปรับปรุงแก้ไข

5. สิ่งที่จะดำเนินการต่อไป

ศัตรู (Enemy)

ในส่วน AI ของศัตรูจะทำการพัฒนาให้สมบูรณ์เท่าที่จะทำได้โดยในส่วนนี้จะรวมไปถึงบอสต่างๆในเกมด้วย

ระบบการต่อสู้ (Combat System)

ในส่วนของการพัฒนาระบบต่อสู้นั้น จะทำการพัฒนาการออกท่าทางการ โจมตีหรือ โดน โจมตีของตัวละคร

- พัฒนาระบบการ โจมตีของตัวละคร(การค์อาวุธ) : โจมตีของตัวละครจะขึ้นอยู่กับอาวุธที่ตัวละครสวมใส่อยู่ ซึ่งสามารถใส่ได้พร้อมกันสูงสุด 5 อาวุธ ซึ่งเมื่อโจมตีจะเป็นการใช้การค์อาวุธแต่ละชิ้นที่สวมใส่อยู่ไปตามลำดับ เมื่อครบการ โจมตีแล้ว หรือ หยุดการ โจมตี จะกลับไปเริ่มต้นที่อาวุธแรกใหม่ ทำให้สามารถสร้างรูปแบบการ โจมตีใหม่ๆได้ตามที่ผู้เล่นต้องการ
- ทำ Template ของอาวุธ : ทำการเพิ่มอาวุธชนิดต่างๆ เข้ามาเพื่อเพิ่มความหลากหลายในการใช้ระบบการ โจมตี
- ทำ Animation ของอาวุธ : ทำการเพิ่มท่าทางการ โจมตีที่แตกต่างกันกันไปของแต่ละอาวุธ
- ทำ Animation การ โจมตีและ โดน โจมตีของตัวละคร : เมื่อตัวละคร โดน โจมตีตัวละครอาจจะมี Animation การ ชะงัก ทำให้ไม่สามารถ โจมตีได้