

การพัฒนาเกม 3D RPG
3D RPG GAME DEVELOPMENT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์(ต่อเนื่อง) คณะวิศวกรรมศาสตร์ ปีการศึกษา 2565
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์(ต่อเนื่อง)

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาเกม 3D RPG

3D RPG GAME DEVELOPMENT

ผู้จัดทำ

นายณัฐพล สุขสมบูรณ์

รหัสนักศึกษา 63015049



อาจารย์ที่ปรึกษา

(อาจารย์ธนัญชัย ตรีภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเกม 3D RPG

นายณัฐพล สุขสมรูป 63015049
อาจารย์รัชัญชัย ตริภาค อาจารย์ที่ปรึกษา
ปีการศึกษา 2565

บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อวิเคราะห์ ออกแบบ และพัฒนาเกม RPG GAME โดยมีการใช้ AI มาปรับสมดุลต่างๆ ภายในเกมให้ผู้เล่นสามารถเล่นเกมได้อย่างสนุกสนาน การพัฒนาจะใช้โปรแกรม Unity ในการพัฒนาและสร้างตัวเกม มีการใช้ Microsoft Visual Studio 2019 ภาษา (C#) ในการออกแบบระบบและการเคลื่อนไหวหรือท่าทางตัวละครกับการเพิ่ม Level จาก Exp ที่ได้รับ ส่วนของ UI ของเกม ใช้โปรแกรม Adobe Photoshop กับ Adobe Illustrator ในการออกแบบหน้าต่างและไอคอนภายในเกม

จากการพัฒนาเกม RPG GAME พบว่ามีระบบที่สามารถทำงานตอบสนองต่อผู้ใช้ระบบได้ เช่น ระบบการเดิน/กระโดด/วิ่ง กับการโจมตีของตัวละคร ระบบคะแนน ระบบการลดค่า HP จากความเสียหายหรือความเสียหายจากการโจมตี ระบบ Level ของตัวละคร สามารถคูณนิแมฟได้ ระบบ AI ภายในเกม ระบบสรุปผลคะแนน ฉาก การเลือกโหมด และบรรยากาศของด่านภายในเกม

3D RPG GAME DEVELOPMENT

Mr. Nattaphon Suksomroop 63015049

Mr. Thananchai Threepak Advisor

Academy Year 2022

Abstract

This project aims to analyze, design and develop RPG GAME by using AI to balance various things. The game allows players to play the game for fun. The development will use Unity to develop and create the game. Microsoft Visual Studio 2019 is used for system design and movement. In the UI of the game, Adobe Photoshop and Adobe Illustrator are used to design the windows. and icons, including graphics within the game.

From the development of RPG GAME games, there must be a system that can support the work or not. System users can set settings such as set/jump/run. with the commander's commander Rate the use of the operating system by using the HP value from the damet or the minutes of the supervisor's meeting. The operating system of alternate levels, crowns, mini-maps, AI systems within the game, analyzing scores, environments, mode selection, and atmosphere of levels within the game.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ 3D RPG GAME DEVELOPMENT สามารถดำเนินการจนประสบความสำเร็จ ลุล่วงได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์และการสนับสนุนอย่างดียิ่งจาก อาจารย์ธัญชัย ตรีภาค อาจารย์ที่ปรึกษา ที่ได้กรุณาให้คำปรึกษา ข้อคิด ข้อเสนอแนะ และปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ จนกระทั่งโครงการนี้สำเร็จเรียบร้อยด้วยดี ผู้จัดทำโครงการขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณคณะอาจารย์ สาขาวิศวกรรมคอมพิวเตอร์(ต่อเนื่อง) คณะวิศวกรรมศาสตร์ทุกท่านที่กรุณาอบรมสั่งสอนให้ความรู้ในการเขียนโปรแกรมเพื่อการพัฒนาาระบบและเรียนรู้การบริหารงานต่างๆ ส่งผลให้ผู้จัดทำสามารถนำความรู้ที่ได้มาประยุกต์ใช้ในการสร้างผลงานสำหรับโครงการเรื่องเกม 3D RPG GAME DEVELOPMENT นี้ให้สำเร็จลุล่วงด้วยดี

ท้ายที่สุด ผู้จัดทำโครงการหวังว่า โครงการฉบับนี้ จะเป็นประโยชน์ให้ต่อผู้ใช้ระบบที่สนใจไม่มากก็น้อย

ณัฐพล

สุขสมรูป

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	IV
สารบัญตาราง	V
สารบัญภาพ	VI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 เป้าหมายของโครงการ	2
1.4 ขอบเขตของโครงการ	2
1.5 แผนการดำเนินงานและระยะเวลาการทำโครงการ	3
1.6 เทคโนโลยีที่เกี่ยวข้อง	6
1.7 ประโยชน์ที่คาดว่าจะได้รับ	6
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	7
2.1 ความรู้ทั่วไปเกี่ยวกับอินเทอร์เน็ตและเกมออนไลน์	7
2.2 แนวคิดพื้นฐานของทฤษฎีเกม	8
2.3 รูปแบบการนินยามเกม	9
2.4 ทฤษฎีการคำนวณเบื้องต้น	10
2.5 อัลกอริทึม	12
2.6 Reinforcement Learning (RL)	15
2.7 การฝึกบอสให้เก่งด้วย RL	17

สารบัญ(ต่อ)

	หน้า
2.8 AI.....	20
บทที่ 3 การออกแบบ.....	23
3.1 Requirement.....	23
3.2 การออกแบบในส่วนของเกม.....	25
3.3 แผนภาพ Use Case Diagram.....	31
3.4 User Interface.....	44
3.5 Class Diagram.....	46
3.6 เครื่องมือที่เกี่ยวข้อง.....	49
บทที่ 4 ผลการดำเนินงาน.....	52
4.1 วิเคราะห์พัฒนาระบบในการเล่นเกม.....	52
4.2 การทดลองการทำงานของตัวเกม.....	55
บทที่ 5 สรุปผล อภิปราย และข้อเสนอแนะ.....	76
5.1 สิ่งที่ทำไปแล้ว.....	76
5.2 ปัญหาและอุปสรรค.....	77
5.3 ประโยชน์ที่คาดว่าจะได้รับ.....	77
5.4 การอภิปรายผล.....	78

สารบัญตาราง

ตาราง	หน้า
1.1 แผนการดำเนินงานภาคการศึกษาที่ 1/2565	3
1.2 แผนการดำเนินงานภาคการศึกษาที่ 2/2565	4
1.3 สรุปฟังก์ชันการทำงานและสิทธิการเข้าใช้งาน เกม 3D RPG	4
2.1 ตัวอย่างการนำเสนอเกมเป่ายิงลูก.....	9
2.2 Q – Table.....	18
3.1 Requirement.....	23



สารบัญรูป

รูป	หน้า
1.1 แผนการดำเนินงาน (Gantt Chat)	5
2.1 ทฤษฎีตรรกศาสตร์ที่เกี่ยวข้องในการประยุกต์ใช้ LUDOCORE.....	11
2.2 A* Algorithm.....	14
2.3 การทำงานของ A*.....	15
2.4 ลำดับการทำงานของ Reinforcement learning (RL).....	16
2.5 ตารางระหว่างState-Action (Q-table)	17
2.6 Classic RL Diagram.....	17
2.7 อัลกอริทึม Info GAIL.....	20
3.1 ตัวละครผู้เล่น Player หญิง(นักธนู).....	26
3.2 ออกแบบแมพทั้ง 3 โหมด ทั้ง Easy, Medium , Hard.....	27
3.3 ตัวละครศัตรูหรือ Enemy Skeleton.....	28
3.4 ตัวละครศัตรูหรือ Enemy NightmareDragon.....	28
3.5 ตัวละคร Boss.....	29
3.6 หาเส้นทางที่สั้นที่สุดโดยการสร้าง Graph ขึ้นเช็คตำแหน่งของ Node.....	30
3.7 วาร์ป (Warp)	30
3.8 Use Case ของหน้าจอ Home Screen.....	32
3.9 Use Case ของหน้าจอ ModeScreen.....	34
3.10 Use Case ของหน้าจอ Game Screen.....	37
3.11 Use Case ของหน้าจอ Score Screen.....	42
3.12 UI Home Screen.....	44
3.13 UI Mode Screen.....	44
3.14 UI Game Screen.....	45
3.15 UI Score Screen.....	46

สารบัญรูป(ต่อ)

รูป	หน้า
3.16 Class Diagram Player.....	46
3.17 Class Diagram Warp.....	47
3.18 Class Diagram UI.....	47
3.19 Class Diagram Enemy.....	48
3.20 เว็บไซต์ Maximo.....	49
3.21 Asset Store ของ Unity.....	49
3.22 Chine machine Camera ของ Registry Unity.....	50
3.23 Machine Learning.....	50
3.24 Training our Player agent using multiple instances.....	51
4.1 Activity Diagram การทำงานของระบบเกม.....	54
4.2 ทดสอบการเคลื่อนที่ของตัวละครและการหมุนมุกกล้อง.....	56
4.3 ทดสอบการโจมตีระยะไกล.....	56
4.4 การโจมตีของ Enemy_Skeleton และ NightmareDragon.....	57
4.5 Skill การโจมตีของ Boss.....	57
4.6 Enemy AI A* PathFinding.....	59
4.7 ระบบ EXP และระบบ Level ของผู้เล่น.....	60
4.8 ทดสอบการทำ Damage ให้กับตัว Enemy.....	61
4.9 การตาย Die ของ Player.....	62
4.10 การตาย Die ของ Enemy.....	62
4.11 Scenes Mode Easy.....	63
4.12 Scenes Mode Medium, Hard.....	64
4.13 Scenes Mode Hard.....	64

สารบัญรูป(ต่อ)

รูป	หน้า
4.14 ทดสอบการวาร์ป.....	65
4.15 ทดสอบการวาร์ปโดยทำให้จุดวาล่องหน.....	65
4.16 การทำ Damage ของ Player.....	66
4.17 การทำ Damage ของ Enemy.....	67
4.18 Action การเคลื่อนไหวตัวละครและ Action อื่นๆ.....	68
4.19 UI Home.....	69
4.20 UI Mode.....	69
4.21 UI Game.....	70
4.22 UI Score.....	71
4.23 การแบ่ง Training Boss หลายๆ Object เพื่อทดสอบผลลัพธ์.....	72
4.24 ผลลัพธ์การ Training ผ่านการ config Anaconda.....	72
4.25 ผลลัพธ์ไฟล์ Result RL.....	73
4.26 Minimap.....	73
4.27 Score.....	74
4.28 ทดสอบ Run บน Web.....	75

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

โลกในยุคปัจจุบัน แทบจะเรียกได้ว่าเกมเป็นส่วนหนึ่งของชีวิตในหมู่เด็ก ๆ รวมทั้งเยาวชน เนื่องจากเกมเป็นสิ่งที่เข้าถึงได้ไม่ยาก โดยแต่ละเกมนั้นมีข้อดีและข้อเสียต่างกัน ซึ่งเกมในปัจจุบันก็มีมากมายหลายรูปแบบ เช่น เกมส่งเสริมทักษะการคำนวณ เกมส่งเสริมทักษะการวางแผน เกมส่งเสริมทักษะการสังเกต เกมส่งเสริมด้านกีฬา เกมส่งเสริมด้านการศึกษา จึงเป็นเรื่องสำคัญที่เราจะต้องเลือกสรรเล่นเกมที่มีคุณภาพ และให้ประโยชน์แก่ผู้เล่นเกม สมาชิกในทีมเล็งเห็นถึงความสำคัญในจุดนี้ จึงมีความคิดที่จะพัฒนาเกมที่สร้างสรรค์ และเป็นประโยชน์กับเยาวชน โดยในโอกาสนี้จะมุ่งเน้นถึงการสร้างสรรค์ และการพัฒนาเกม ผ่านเกมคอมพิวเตอร์ในรูปแบบของการพัฒนาการ 3D แนว RPG

เนื่องจากการเล่นเกมเป็นที่นิยมอย่างมากในปัจจุบัน โครงการนี้จึงได้พัฒนาขึ้นจากความสนใจในการเล่นเกมนแนว Role-Playing Game หรือ RPG คือแนวเกมที่เราจะได้สวมบทบาทเป็นตัวละครตัวหนึ่ง ซึ่งผู้จัดทำได้กำหนดให้ผู้เล่นสวมบทบาทเป็นนักธนูที่มีลักษณะต้องเอาชีวิตรอดและหาทางชนะศัตรูให้ได้ด้วยการฝึกทักษะการโจมตี การเรียนรู้ที่จะเอาชนะเกม โดยพัฒนาเกมที่สร้างสรรค์ผู้เล่นสามารถเลือกโหมดที่เล่นตามทักษะที่มีและสามารถเอาชนะได้ตั้งแต่ผู้เริ่มต้นการเล่นจนถึงผู้ที่มีทักษะการเล่นระดับสูงที่เอาชนะเกมอย่างง่ายดายเกินไปไม่มีความน่าตื่นเต้นจึงได้พัฒนาเกมนี้มาจัดทำเป็นโครงการให้ศัตรูสามารถเรียนรู้การชนะด้วยตัวเองได้ และมุ่งเน้นให้ผู้เล่นมีความสุขสนุกสนานในการเล่น ลดความเครียด ใช้ปัญญาในการวิเคราะห์สถานการณ์ต่างๆได้

โครงการนี้จึงนำเทคโนโลยีการออกแบบเกมและความรู้ที่อาจารย์สอน มาใช้ในการพัฒนาเกม 3D แนว RPG เพื่อพัฒนาเกมแนวนี้ให้มีความหลากหลายยิ่งขึ้น โดยแบ่งเป็น 3 โหมดให้เลือกเล่นคือ

- 1) โหมดง่ายสำหรับผู้เล่นเริ่มต้น
- 2) โหมดปานกลางสำหรับผู้เล่นที่เริ่มมีทักษะการเล่นเอาชนะ
- 3) โหมดยากสำหรับผู้เล่นที่ชอบความท้าทายจากการที่ Boss ชนะยากจากการที่สามารถเรียนรู้ตัวเองที่จะชนะผู้เล่น

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อออกแบบและพัฒนาเกม 3D RPG Development
- 2) ศึกษารูปแบบการพัฒนาเกม 3D RPG Development
- 3) เพื่อศึกษาการออกแบบและพัฒนาเกมผ่าน โปรแกรม Unity

1.3 เป้าหมายของโครงการ

- 1) ได้ออกแบบและพัฒนาเกม 3D RPG Development
- 2) ได้ศึกษารูปแบบการพัฒนาเกม 3D RPG Development
- 3) ได้ศึกษาการออกแบบและพัฒนาเกมผ่าน โปรแกรม Unity

1.4 ขอบเขตของโครงการ

เกมสวมบทบาท หรือเกม RPG (Role – Playing game : RPG) คือเกมประเภทหนึ่งที่มีผู้เล่นสมมุติ รับบทเป็นตัวละครหนึ่งในเกม โดยเล่นตามกฎกติกาของเกมผ่านการป้อนคำสั่งและเลือกเงื่อนไขที่เกม กำหนดมาโดยผลลัพธ์ที่เกิดจะแตกต่างกัน ตามเงื่อนไขที่เลือก โดยเกมอาจจะเป็นลักษณะ การเล่นโดย เขียนในกระดานวิดีโอเกม หรือคอมพิวเตอร์เกมก็ได้

เกมที่จะพัฒนานี้ก็จะเป็นเกมรูปแบบ Animation 3D มุมมองบุคคลที่สาม ควบคุมด้วยเมาส์ละ คีย์บอร์ด ทางเดินไปยังเป้าหมายต่าง ๆ ซึ่งในแต่ละด่าน จะมีลักษณะภูมิภาคน่าไม่เหมือนกัน จะแบ่ง ออกเป็นทั้งหมด 3 ด่าน Easy Normal และ Hard เกมจะเป็นในรูปแบบโจรสลัด และมีเป้าหมายของ ภารกิจที่ตัวเกมจะให้ออก เมื่อถึงเป้าหมายที่ตัวเกมตั้งไว้ เกมจะทำการสรุปคะแนนที่ได้ภายในเกม และ ไปด่านต่อไป เกมจะจบเมื่อดำเนินการเล่นครบทั้ง 3 ด่านถึงจะบรรลุเป้าหมาย

1.5 แผนการดำเนินงานและระยะเวลาการทำงาน

ตาราง 1.1 แผนการดำเนินงานภาคการศึกษาที่ 1/2565

ลำดับ	ขั้นตอนการดำเนินงาน	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	กำหนดหัวข้อและกำหนดขอบเขตของโครงการ	■	■	■													
2	ศึกษาและค้นคว้างานวิจัยที่เกี่ยวข้องและศึกษาเครื่องมือเครื่องใช้ของ Unity	■	■	■						■	■	■					
3	เริ่มพัฒนาระบบ			■	■	■	■	■									
4	วิเคราะห์รูปแบบการสร้างระบบและออกแบบเกม			■	■	■	■	■				■	■				
5	ทดสอบระบบ			■	■	■	■	■				■	■	■	■		
5	ปรับปรุงและแก้ไขประสิทธิภาพระบบเกม			■	■	■	■	■				■	■	■	■		
6	สรุปผลระบบโครงการทั้งหมด													■	■	■	
7	จัดทำรายงานโครงการ 1									■	■	■	■	■	■	■	■

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 1.2 แผนการดำเนินงานภาคการศึกษาที่ 2/2565

ลำดับ	ขั้นตอนการดำเนินงาน	มกราคม				กุมภาพันธ์				มีนาคม				เมษายน			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	พัฒนาระบบ																
2	ทดสอบโปรแกรมและ การใช้งานบนเว็บ																
3	ตรวจสอบปัญหาและ แก้ไขปรับปรุง																
4	จัดทำคู่มือการใช้งานเกม																
5	จัดทำรายงาน โครงการงาน 2																
5	ส่งไฟล์นำเสนอสำหรับ สอบโครงการ																

ตาราง 1.3 สรุปฟังก์ชันการทำงานและสิทธิการใช้งาน เกม 3D RPG

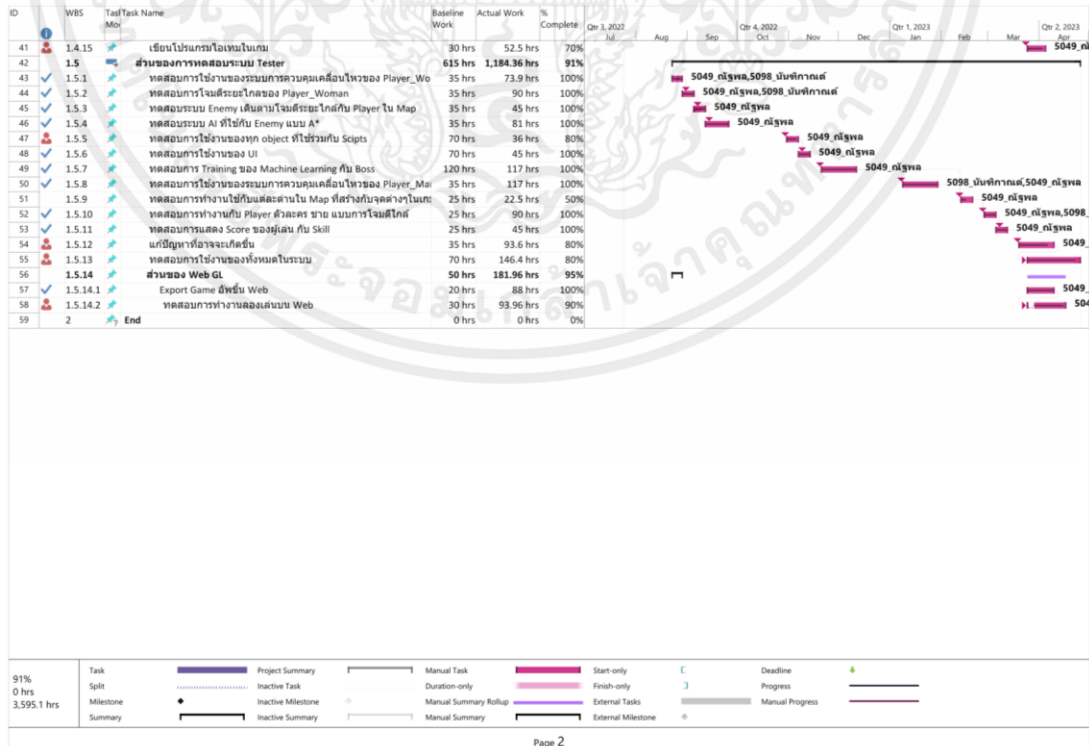
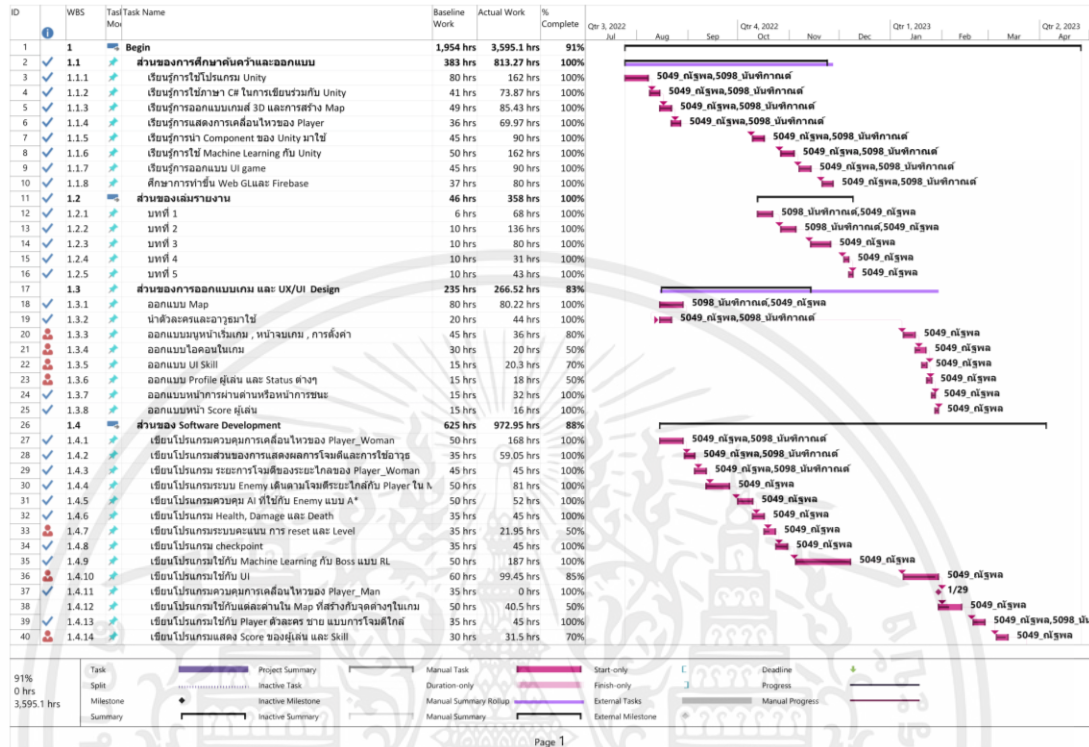
ฟังก์ชันการทำงานของระบบ	กลุ่มผู้มีสิทธิใช้งาน	
	ผู้ดูแลระบบ	ผู้เล่น
ระบบด้านภายในเกม	A	U
ระบบท่าทางตัวละคร	A	U
ระบบเลเวลตัวละคร	A	U
การควบคุมตัวละคร	A	U
เสียงประกอบเกม	A	U

หมายเหตุ สัญลักษณ์ หมายถึง A สามารถแก้ไขได้

หมายถึง U สามารถใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 1.1 แผนการดำเนินงาน (Gantt Chat)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 เทคโนโลยีที่เกี่ยวข้อง

1.6.1 Software

1.6.2.1 โปรแกรม Unity ใช้ในการออกแบบและจัดการพัฒนาสร้างตัวเกม

1.6.2.2 Microsoft Visual Studio 2019 ใช้ในการเขียน Script ควบคุมต่าง ๆ

1.6.2.3 PyTorch ใช้ Python API ในการฝึกตัวแทน Boss ด้วย RL

1.6.2.4 Anaconda ใช้ในการ Config และ Install เครื่องมือในการเรียกใช้งานต่างๆ

1.7 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้ฝึกทักษะในการคิดระบบเกม เรียนรู้กลไกการทำงานของเกม
- 2) สามารถนำความรู้ไปใช้ได้ตามวัตถุประสงค์
- 3) สามารถนำไปประยุกต์ต่อยอดการพัฒนาเกมในแพลตฟอร์มอื่นได้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาครั้งนี้ผู้วิจัยมุ่งศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง และนำเสนอสาระสำคัญดังนี้คือ ศึกษาพฤติกรรมการเล่นเกมออนไลน์ของนักศึกษาสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จึงได้ศึกษาแนวคิดและทฤษฎีที่เกี่ยวข้อง ดังนี้

2.1 ความรู้ทั่วไปเกี่ยวกับอินเทอร์เน็ตและเกมออนไลน์

2.1.1 ความรู้ทั่วไปเกี่ยวกับอินเทอร์เน็ต

ความรู้ทั่วไปเกี่ยวกับอินเทอร์เน็ตได้ถูกนำมาใช้งานแพร่หลาย และเป็นที่ยอมรับมากในภาคธุรกิจและผู้บริโภคทั่วไปเป็นการสร้างโอกาสใหม่ๆ ของการทำธุรกิจ หรือบริการ และเข้าสู่ลูกค้ากลุ่มใหม่ และทำให้การทำธุรกิจกับลูกค้ากลุ่มเดิมเป็นไปได้อย่างรวดเร็ว สื่ออินเทอร์เน็ตในปัจจุบันจึงเป็นสื่อที่มีศักยภาพ และแตกต่างจากสื่อที่เคยมีมาเป็นอย่างมาก นับแต่การนำเสนอที่ไม่มีข้อจำกัดระบบการสืบค้นข้อมูลที่ถูกจัดเก็บในรูปของภาษา ที่สามารถสื่อสารบนระบบอินเทอร์เน็ตให้เชื่อมโยงถึงกันได้

2.1.2 ความรู้ทั่วไปเกี่ยวกับเกม

เกมเป็นลักษณะของกิจกรรมของมนุษย์เพื่อประโยชน์อย่างใดอย่างหนึ่ง เช่น เพื่อความสนุกสนานบันเทิง เพื่อฝึกทักษะ และเพื่อการเรียนรู้ เป็นต้น และในบางครั้งอาจใช้เพื่อประโยชน์ทางการศึกษา เกมประกอบด้วยเป้าหมาย กฎเกณฑ์ การแข่งขันและปฏิสัมพันธ์ เกมมักจะเป็นการแข่งขันทางจิตใจหรือด้านร่างกาย หรือทั้งสองอย่างรวมกัน ซึ่งส่งผลให้เกิดพัฒนาการของทักษะ ใช้เป็นรูปแบบของการออกกำลังกาย หรือการศึกษา บทบาทสมมุติและจิตศาสตร์ เป็นต้น

2.1.3 ความรู้ทั่วไปเกี่ยวกับเกมออนไลน์

ผู้วิจัยได้ทำการทบทวนทฤษฎี เอกสาร และงานวิจัยที่เกี่ยวข้องกับเกมคอมพิวเตอร์ออนไลน์ พฤติกรรมการติดเกมคอมพิวเตอร์ออนไลน์ในเด็กและเยาวชนของประเทศ การใช้เทคโนโลยีในการป้องกันและแก้ไขปัญหาการติดเกมคอมพิวเตอร์ออนไลน์ในเด็กและเยาวชน ตลอดจนการบริหารจัดการเทคโนโลยีดังกล่าว ทั้งในและต่างประเทศ เพื่อนำมาสร้างเป็นกรอบแนวคิดการศึกษาวิจัยในการสร้างแบบจำลองและตัวชี้วัด (Indicators) เพื่อติดตามพฤติกรรมและแปรผลสถานะเสี่ยงการติดเกมใน

เด็กและเยาวชนของประเทศไทย ในเบื้องต้นนี้ ผู้วิจัยได้ทำการศึกษาทบทวนวรรณกรรมเพื่อทำความเข้าใจถึงนิยามและรูปแบบต่าง ๆ ของเกม เพื่อใช้เป็นความรู้พื้นฐานในการศึกษาในมิติอื่น ๆ ที่เกี่ยวข้องกับเกมคอมพิวเตอร์ออนไลน์ต่อไป โดยมีรายละเอียด ดังนี้ “เกม” เป็นคำนามและมีความหมายตามพจนานุกรมฉบับราชบัณฑิตยสถาน ปี พ.ศ.2542 ว่า “การแข่งขันที่มีกติกากำหนด เช่น เกมกีฬา, การเล่นเพื่อความสนุก เช่น เกมคอมพิวเตอร์, การแสดงเพื่อสาธิตกิจกรรม เช่น เกมการบริหาร, โดยปริยายหมายถึงการแสดงที่ใช้กลวิธีหรือเล่ห์เหลี่ยมเพื่อหักล้างกัน เช่น เกมการเมือง, ลักษณะนามเรียกการแข่งขันหรือการเล่นเกมที่จบลงด้วยการแพ้ชนะกันครั้งหนึ่ง ๆ เช่น เล่นแบดมินตัน 3 เกม (อ. Game)” ทั้งนี้คำว่าเกมซึ่งเป็นคำทับศัพท์มาจากภาษาอังกฤษ (Game) ซึ่งหมายถึง เกมอิเล็กทรอนิกส์ (Electronic) ที่เล่นโดยภาพเคลื่อนไหวบนฉาก (Screen) หรือจอ (Monitor) หรือออนไลน์เกม (Online Game) ที่มีความหมายถึง เกมที่ถูกเล่น โดยเชื่อมต่อกับอินเทอร์เน็ต (Internet) เป็นต้น

2.2 แนวคิดพื้นฐานของทฤษฎีเกม

ทฤษฎีเกม สาขาคณิตศาสตร์ประยุกต์ที่มีเครื่องมือในการวิเคราะห์สถานการณ์ที่ฝ่ายต่างๆ เรียกว่าผู้เล่น ตัดสินใจพึ่งพากัน การพึ่งพาซึ่งกันและกันนี้ทำให้ผู้เล่นแต่ละคนต้องพิจารณาการตัดสินใจหรือกลยุทธ์ที่เป็นไปได้ของผู้เล่นคนอื่นๆ ในการกำหนดกลยุทธ์ วิธีแก้ปัญหของเกมจะอธิบายถึงการตัดสินใจที่เหมาะสมที่สุดของผู้เล่น ซึ่งอาจมีความสนใจคล้ายกัน ต่อต้าน หรือผสมกัน และผลลัพธ์ที่อาจเป็นผลมาจากการตัดสินใจเหล่านี้ แม้ว่าทฤษฎีเกมจะถูกนำมาใช้ในการวิเคราะห์เกมในห้องนั่งเล่น แต่การใช้งานของมันนั้นกว้างกว่ามาก อันที่จริง เดิมทีทฤษฎีเกมได้รับการพัฒนาโดยนักคณิตศาสตร์ชาวอเมริกันเชื้อสายฮังการี จอห์น ฟอน นอยมันน์ และเพื่อนร่วมงานที่มหาวิทยาลัยพรินซ์ตัน ออสการ์ มอร์เกนสเติร์น นักเศรษฐศาสตร์ชาวอเมริกันเชื้อสายเยอรมัน เพื่อแก้ปัญหาทางเศรษฐศาสตร์ ในหนังสือของพวกเขา The Theory of Games and Economic Behavior (1944) ฟอน นอยมันน์และมอร์เกนสเติร์นยืนยันว่าคณิตศาสตร์ที่พัฒนาขึ้นสำหรับวิทยาศาสตร์กายภาพ ซึ่งอธิบายถึงการทำงานของธรรมชาติที่ไม่สนใจ เป็นแบบอย่างที่ไม่ดีสำหรับเศรษฐศาสตร์ พวกเขาสังเกตเห็นว่าเศรษฐศาสตร์ก็เหมือนกับเกม ซึ่งผู้เล่นต่างคาดเดาการเคลื่อนไหวของกันและกัน ดังนั้นจึงต้องใช้คณิตศาสตร์รูปแบบใหม่ ซึ่งพวกเขาเรียกว่าทฤษฎีเกม (ซึ่งนี้อาจเป็นชื่อเรียกที่ผิด ทฤษฎีเกมโดยทั่วไปไม่ได้แบ่งปันความสนุกหรือความไร้สาระที่เกี่ยวข้องกับเกม)

ทฤษฎีเกมถูกนำไปใช้กับสถานการณ์ที่หลากหลายซึ่งตัวเลือกของผู้เล่นโต้ตอบกันเพื่อส่งผลกระทบต่อผลลัพธ์ ในการเน้นย้ำถึงแง่มุมเชิงกลยุทธ์ของการตัดสินใจ หรือแง่มุมที่ควบคุมโดยผู้เล่นแทนที่จะเป็นความบังเอิญ ทฤษฎีทั้งสองเสริมและนอกเหนือไปจากทฤษฎีดั้งเดิมของความน่าจะเป็น มีการใช้

ตัวอย่างเช่น เพื่อกำหนดแนวร่วมทางการเมืองหรือกลุ่มธุรกิจที่มีแนวโน้มว่าจะก่อตัวขึ้น ราคาที่เหมาะสมที่สุดที่จะขายผลิตภัณฑ์หรือบริการท่ามกลางการแข่งขัน อำนาจของผู้มีสิทธิเลือกตั้งหรือกลุ่มผู้มีสิทธิเลือกตั้ง เลือกคณะกรรมการตัดสิน สถานที่ที่ดีที่สุดสำหรับโรงงานผลิต และพฤติกรรมของสัตว์และพืชบางชนิดในการต่อสู้เพื่อความอยู่รอด มันถูกใช้เพื่อทำทฤษฎีความถูกต้องตามกฎหมายของระบบการลงคะแนนเสียงบางระบบ

2.3 รูปแบบการนิยามเกม

เกมรูปแบบกลยุทธ์ (strategic-form game) หรือเกมรูปแบบปกติ (normal-form game) ประกอบไปด้วยการระบุผู้เล่นภายในเกม ทางเลือกของผู้เล่นแต่ละฝ่าย เรียกในทางทฤษฎีเกมว่ากลยุทธ์ และฟังก์ชันอรรถประโยชน์ของผู้เล่นแต่ละฝ่าย ในกรณีที่มีผู้เล่นสองฝ่าย และแต่ละฝ่ายมีทางเลือกจำนวนจำกัด เกมนี้สามารถเขียนออกมาได้ในรูปของตารางโดยให้แต่ละแถวในตารางหมายถึงทางเลือกของผู้เล่นฝ่ายหนึ่ง และแต่ละสดมภ์หมายถึงทางเลือกของผู้เล่นอีกฝ่ายหนึ่ง ช่องของตารางแต่ละช่องระบุอรรถประโยชน์ของผู้เล่นสองฝ่ายในแต่ละกรณี ดังตาราง 2.1

ตารางที่ 2.1 ตัวอย่างการนำเสนอเกมเป่ายิงฉุบ

	ก้อน	กรรไกร	กระดาษ
ก้อน	0,0	1,-1	-1,1
กรรไกร	-1,1	0,0	1,-1
กระดาษ	1,-1	-1,1	0,0

โดยทั่วไปแล้ว จำนวนทางเลือกของผู้เล่นไม่จำเป็นต้องมีจำนวนจำกัด (ตัวอย่างกรณีที่ผู้เล่นมีทางเลือกไม่จำกัดคือ ผู้ขายสินค้าสามารถตั้งราคาขายสินค้าเป็นตัวเลขใดๆ ก็ได้) หากว่าทางเลือกของผู้เล่นทุกฝ่ายมีจำนวนจำกัด ทางเลือกในกรณีนี้จะเรียกว่าเป็นกลยุทธ์แท้ เกมกลยุทธ์แท้สามารถขยายให้ผู้เล่นสามารถเลือกกำหนดความน่าจะเป็นที่จะสุ่มเลือกทางเลือกแต่ละทาง เรียกว่ากลยุทธ์ผสม ตัวอย่างเช่น ในเกมเป่ายิงฉุบข้างต้นจากบทความ (จอห์น ฟอน นอยมันน์, 1923) ได้เขียนถึงการใช้กลยุทธ์ผสมว่า "สามัญสำนึกจะบอกได้ว่าวิธีที่ดีที่สุดที่จะเล่นเกมนี้คือการเลือกทางเลือกทั้งสามทางด้วยความน่าจะเป็นแต่ละทางเท่ากับ $\frac{1}{3}$

(2.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ทฤษฎีการคำนวณเบื้องต้น

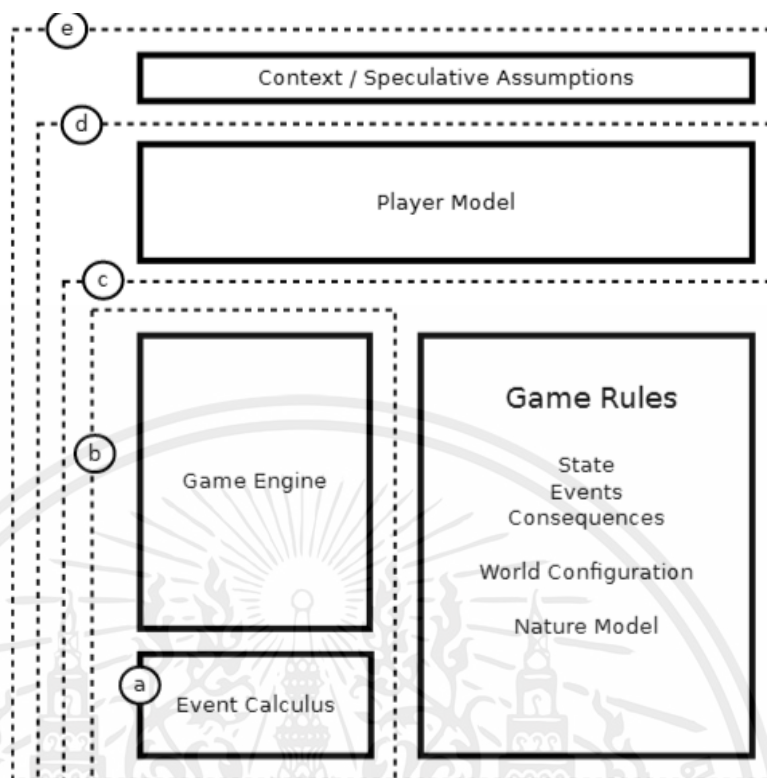
ทฤษฎีเกมมุ่งเข้าไปที่การเพิ่มประสิทธิภาพผลประโยชน์สูงสุดของหลาย ๆ วิชาด้วยข้อจำกัดบางอย่างภายใต้สถานการณ์เฉพาะ ในระหว่างเกม บุคคลต่าง ๆ อาศัยข้อมูลของตนเอง และตามลำดับหรือพร้อมกัน สร้างกลยุทธ์ของตนเองครั้งเดียวหรือซ้ำ ๆ เพื่อกำหนดกลยุทธ์ที่เหมาะสมที่สุดเพื่อเล่นเกมจริง โดยทั่วไปแล้ว เกมที่สมบูรณ์ประกอบด้วยสามองค์ประกอบ: ผู้เล่น กลยุทธ์ และผลตอบแทน (เรียกอีกอย่างว่า payoff function) ผู้เล่นหรือบุคคลในเกมที่สามารถเลือกกลยุทธ์และตัดสินใจได้

นิยามของเกมรูปแบบกลยุทธ์สามารถเขียนได้ว่า เกมรูปแบบกลยุทธ์ดังสมการ 2.1

$$\text{เซตผู้เล่น } N = \{1, 2, \dots, n\} \text{ ทฤษฎีเซตมีจุดเริ่มต้นจาก (เกออร์ค คันทอร์, 1874)} \quad (2.1)$$

สูตรข้างต้นแสดงถึงชุดที่เกิดขึ้นจากการมีส่วนร่วมของผู้เล่น N ในกระบวนการเกม กลยุทธ์เป็นองค์ประกอบที่สำคัญในเกม และสามารถเลือกความเป็นไปได้ของกลยุทธ์และจำนวนกลยุทธ์โดยผู้เล่นที่รวบรวมข้อมูลในกระบวนการและทรัพยากรของเกม

ในโลกของเกม การกำหนดค่าของเอนทิตีภายในโลกของเกม (เช่น ค่าโคงระดับ) และแบบจำลองเมื่อสิ่งต่าง ๆ เกิดขึ้นในโลกของเกมโดยปราศจากการแทรกแซงของผู้เล่น (เกิดจาก "ธรรมชาติ") แม้ว่าสิ่งนี้จะระบุโลกของเกมโดยสมบูรณ์ แต่แบบจำลองของผู้เล่นยังเพิ่มข้อสันนิษฐานเกี่ยวกับประเภทของการกระทำที่ผู้เล่นสามารถกระทำได้ ซึ่งผสมผสานกัน—ไม่ว่าจะเกิดจากข้อจำกัดจริง (เช่น หากผู้เขียนมีในใจถึงการแมปอินพุต 4 ซึ่งจะทำให้การกระทำบางอย่างเป็นไปได้ที่จะดำเนินการพร้อมกัน) หรือเนื่องจากความปรารถนาที่จะตรวจสอบพฤติกรรมบางอย่างของผู้เล่น สุดท้ายสามารถเพิ่มสมมติฐานเชิงคาดเดาที่จำกัดร่องรอยที่สร้างขึ้นตามสถานการณ์บางประเภทที่ผู้เขียนต้องการตรวจสอบ



รูป 2.1 ทฤษฎีตรรกศาสตร์ที่เกี่ยวข้องในการประยุกต์ใช้ LUDOCORE

จากรูป 2.1 บทความของ Adam M. Smith, Mark J. Nelson, Michael Mateas (2010) กล่าวว่า Ludocore เป็น "เอ็นจินเกม" แบบลอจิกัล ซึ่งเชื่อมโยงกฎของเกมตามที่ผู้ออกแบบเกมให้เหตุผลกับตรรกะทางการที่ใช้โดยเครื่องมือให้เหตุผลอัตโนมัติใน AI ความท้าทายหลักในการออกแบบบริดจ์นี้คือการออกแบบการนำเสนอที่กระชับ ปลอดภัย และยืดหยุ่น ซึ่งเข้ากันได้กับความหมายของเกมที่ไม่เคลลอจิกัลสร้างขึ้นด้วยเอ็นจินของเราตั้งใจที่จะนำเสนอ สร้างจากแคลคูลัสเหตุการณ์ พิธีการสำหรับการให้เหตุผลเกี่ยวกับสถานะและเหตุการณ์เมื่อเวลาผ่านไป และชุดของโครงสร้างและสำนวนทั่วไปที่ใช้ในเกมการสร้างแบบจำลอง เรายังเสนอเครื่องมือที่สามารถสร้างร่องรอยการเล่นเกมที่แสดงพฤติกรรมแบบไดนามิกของเกม รองรับการสร้างแบบจำลองส่วนเพิ่มของผู้เล่นและบุคคลที่ไม่ใช่ผู้เล่นในโลกของเกม การปรับเปลี่ยนกฎของเกมโดยไม่มี การเปลี่ยนแปลงที่ไม่ใช่ในท้องถิ่น และการสืบค้นข้อมูลเชิงเวลาและเชิงโครงสร้างในเชิงสำรวจ

2.5 อัลกอริทึม

2.5.1 อัลกอริทึมคืออะไร

อัลกอริทึมเป็นขั้นตอนที่ใช้ในการแก้ปัญหาหรือดำเนินการคำนวณ อัลกอริทึมทำหน้าที่เป็นรายการคำสั่งที่ถูกต้องซึ่งดำเนินการตามที่ระบุทีละขั้นตอนในรูทึนที่ใช้ฮาร์ดแวร์หรือซอฟต์แวร์ อัลกอริทึมถูกนำมาใช้กันอย่างแพร่หลายในทุกด้านของไอที ในวิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ อัลกอริทึมมักหมายถึงขั้นตอนเล็กๆ ที่แก้ปัญหาที่เกิดขึ้น อัลกอริทึมยังใช้เป็นข้อกำหนดสำหรับการประมวลผลข้อมูลและมีบทบาทสำคัญในระบบอัตโนมัติ อัลกอริทึมสามารถใช้สำหรับการเรียงลำดับชุดตัวเลขหรือสำหรับงานที่ซับซ้อนมากขึ้น เช่น การแนะนำเนื้อหาของผู้ใช้บนโซเชียลมีเดีย โดยทั่วไปแล้วอัลกอริทึมจะเริ่มต้นด้วยอินพุตเริ่มต้นและคำสั่งที่อธิบายการคำนวณเฉพาะ เมื่อดำเนินการคำนวณ กระบวนการจะสร้างผลลัพธ์

2.5.2 การทำงานของอัลกอริทึม

อัลกอริทึมสามารถแสดงเป็นภาษาธรรมชาติ ภาษาโปรแกรม รหัสเทียม ฟังงาน และตารางควบคุม ส่วนภาษาที่เป็นธรรมชาตินั้นหายาก เนื่องจากมีความกำกวมมากกว่า ภาษาโปรแกรมมักใช้สำหรับแสดงอัลกอริทึมที่ดำเนินการ โดยคอมพิวเตอร์ อัลกอริทึมใช้อินพุตเริ่มต้นพร้อมกับชุดคำสั่งข้อมูลเข้าเป็นข้อมูลเริ่มต้นที่จำเป็นในการตัดสินใจและสามารถแสดงในรูปแบบของตัวเลขหรือคำข้อมูลอินพุตจะถูกส่งผ่านชุดคำสั่งหรือการคำนวณ ซึ่งอาจรวมถึงกระบวนการทางคณิตศาสตร์และการตัดสินใจ เอาต์พุตเป็นขั้นตอนสุดท้ายในอัลกอริทึม และโดยปกติจะแสดงเป็นข้อมูลเพิ่มเติม ตัวอย่างเช่น อัลกอริทึมการค้นหาใช้คำค้นหาเป็นอินพุตและเรียกใช้ผ่านชุดคำสั่งสำหรับการค้นหาผ่านฐานข้อมูลสำหรับรายการที่เกี่ยวข้องกับคำค้นหา ซอฟต์แวร์การทำงานอัตโนมัติทำหน้าที่เป็นอีกตัวอย่างหนึ่งของอัลกอริทึม เนื่องจากการทำงานอัตโนมัติเป็นไปตามชุดของกฎในการทำงานให้เสร็จสิ้น อัลกอริทึมจำนวนมากประกอบกันเป็นซอฟต์แวร์การทำงานอัตโนมัติ และทั้งหมดทำงานเพื่อทำให้กระบวนการที่กำหนดเป็นไปโดยอัตโนมัติ

2.5.3 ประเภทของอัลกอริทึม

อัลกอริทึมมีหลายประเภท ซึ่งทั้งหมดออกแบบมาเพื่อทำงานที่แตกต่างกัน ตัวอย่างเช่น อัลกอริทึมดำเนินการต่อไปนี้

1. Search engine algorithm. อัลกอริทึมนี้ใช้สตริงการค้นหาของคีย์เวิร์ดและ Operator เป็นข้อมูลเข้า ค้นหาเว็บเพจที่เกี่ยวข้องในฐานข้อมูลที่เกี่ยวข้อง และส่งกลับผลลัพธ์

2. Encryption algorithm อัลกอริทึมการคำนวณนี้จะแปลงข้อมูลตามการกระทำที่ระบุเพื่อปกป้องข้อมูล อัลกอริทึมที่นิยมใช้กัน เช่น Data Encryption Standard ใช้คีย์เดียวกันในการเข้ารหัสและถอดรหัสข้อมูล トラบเท่าที่อัลกอริทึมนั้นซับซ้อนเพียงพอ จะไม่มีใครที่ไม่มีคีย์ที่สามารถถอดรหัสข้อมูลได้
3. Greedy algorithm อัลกอริทึมนี้ช่วยแก้ปัญหาการปรับให้เหมาะสม โดยการค้นหาโซลูชันที่เหมาะสมที่สุดในเครื่อง โดยหวังว่าจะเป็นโซลูชันที่ดีที่สุดในระดับโลก อย่างไรก็ตาม มันไม่ได้รับประกันวิธีแก้ปัญหาที่ดีที่สุด
4. Recursive algorithm. อัลกอริทึมนี้เรียกตัวเองซ้ำๆ จนกว่าจะแก้ปัญหาได้ อัลกอริทึมแบบเรียกซ้ำจะเรียกตัวเองด้วยค่าที่น้อยลงทุกครั้งที่เราเรียกใช้ฟังก์ชันแบบเรียกซ้ำ
5. Backtracking algorithm อัลกอริทึมนี้ค้นหาวิธีแก้ไขปัญหาคำหนดด้วยวิธีที่ละขั้นตอนและแก้ปัญหาทีละส่วน
6. Divide-and-conquer algorithm อัลกอริทึมทั่วไปนี้แบ่งออกเป็นสองส่วน ส่วนหนึ่งแบ่งปัญหาออกเป็นปัญหาย่อยๆ ส่วนที่สองแก้ปัญหาเหล่านี้แล้วรวมเข้าด้วยกันเพื่อสร้างวิธีแก้ปัญหา
7. Dynamic programming algorithm. อัลกอริทึมนี้แก้ปัญหาโดยแบ่งเป็นปัญหาย่อย ผลลัพธ์จะถูกเก็บไว้เพื่อใช้สำหรับปัญหาที่เกี่ยวข้องในอนาคต
8. Brute-force algorithm อัลกอริทึมนี้จะวนซ้ำวิธีแก้ปัญหาที่เป็นไปได้ทั้งหมด ค้นหาวิธีแก้ปัญหาอย่างน้อยหนึ่งรายการสำหรับฟังก์ชันหนึ่งๆ
9. Sorting algorithm อัลกอริทึมการเรียงลำดับใช้เพื่อจัดเรียงโครงสร้างข้อมูลใหม่ตามตัวดำเนินการเปรียบเทียบ ซึ่งใช้ในการตัดสินใจลำดับใหม่สำหรับข้อมูล
10. Hashing algorithm อัลกอริทึมนี้ใช้ข้อมูลและแปลงเป็นข้อความที่เหมือนกันพร้อมการแฮช
11. Randomized algorithm อัลกอริทึมนี้ใช้ข้อมูลและแปลงเป็นสิ่งที่เหมือนกันพร้อมแฮช

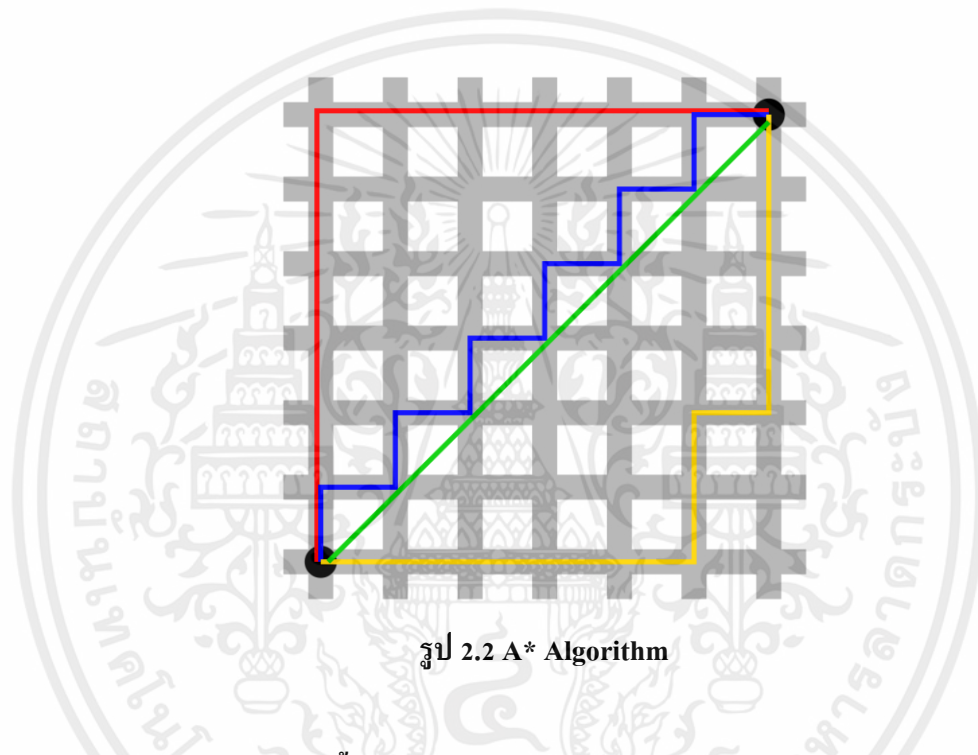
2.5.3 A* Algorithm

2.5.3.1 A* Algorithm

เป็นอัลกอริทึมที่มีประโยชน์ซึ่งมักใช้สำหรับการสำรวจแผนที่เพื่อค้นหาเส้นทางที่สั้นที่สุด เริ่มแรก A* ได้รับการออกแบบให้เป็นปัญหาการผ่านกราฟ เพื่อช่วยสร้างหุ่นยนต์ที่สามารถค้นหาเส้นทางของตัวเองได้ มันยังคงเป็นอัลกอริทึมที่ได้รับความนิยมอย่างกว้างขวางสำหรับการสำรวจกราฟ โดยจะค้นหาเส้นทางที่สั้นกว่าก่อน จึงทำให้เป็นอัลกอริทึมที่เหมาะสมและสมบูรณ์ที่สุด อัลกอริทึมที่ดีที่สุดจะค้นหาผลลัพธ์ที่มีต้นทุนน้อยที่สุดสำหรับปัญหาหนึ่งๆ ในขณะที่อัลกอริทึมที่สมบูรณ์จะค้นหาผลลัพธ์ที่เป็นไปได้ทั้งหมดของปัญหา ในอีกมุมหนึ่งที่ทำให้ A* มีประสิทธิภาพมากคือการใช้กราฟ

ถ่วงน้ำหนักในการนำไปใช้งาน กราฟถ่วงน้ำหนักใช้ตัวเลขเพื่อแสดงต้นทุนของการดำเนินแต่ละเส้นทางหรือแนวทางการดำเนินการ ซึ่งหมายความว่าอัลกอริทึมสามารถใช้เส้นทางโดยมีค่าใช้จ่ายน้อยที่สุด และค้นหาเส้นทางที่ดีที่สุดในแง่ของระยะทางและเวลา

ข้อเสียที่สำคัญของอัลกอริทึมคือความซับซ้อนของพื้นที่และเวลา ต้องใช้พื้นที่จำนวนมากในการจัดเก็บเส้นทางที่เป็นไปได้ทั้งหมดและใช้เวลามากในการค้นหาดังรูป 2.2 แสดงถึงเส้นทางสีเขียวคือเส้นทางที่สั้นที่สุดของวงจุดสีดำดังรูป 2.2



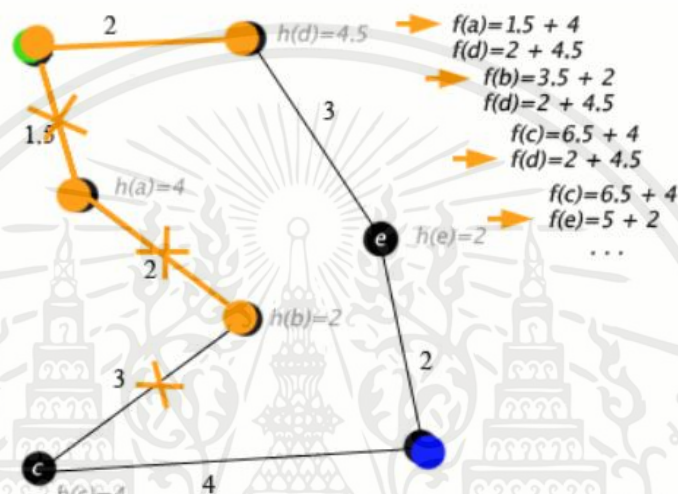
รูป 2.2 A* Algorithm

2.5.3.2 แนวคิดพื้นฐานของ A* Algorithm

อัลกอริทึมแบบฮิวริสติกยอมเสียสละความเหมาะสมด้วยความแม่นยำและความรวดเร็ว เพื่อแก้ปัญหาได้เร็วและมีประสิทธิภาพมากขึ้นกราฟทั้งหมดมีโหนดหรือจุดที่ต่างกันซึ่งอัลกอริทึมต้องใช้เพื่อไปให้ถึงโหนดสุดท้าย เส้นทางระหว่างโหนดเหล่านี้ล้วนมีค่าเป็นตัวเลข ซึ่งถือว่าเป็นน้ำหนักของเส้นทาง ผลรวมของเส้นทางทั้งหมดตามขวางจะให้ราคาของเส้นทางนั้นในขั้นต้น อัลกอริทึมจะคำนวณค่าใช้จ่ายสำหรับ โหนดที่อยู่ใกล้เคียงทั้งหมด n และเลือกโหนดที่มีค่าใช้จ่ายน้อยที่สุด กระบวนการนี้ทำซ้ำจนกว่าจะไม่สามารถเลือกโหนดใหม่ได้ และเส้นทางทั้งหมดถูกข้ามไปแล้ว จากนั้นคุณควรพิจารณาเส้นทางที่ดีที่สุดในหมู่พวกเขา ถ้า $f(n)$ แทนต้นทุนขั้นสุดท้าย จะแสดงเป็น

$f(n) = g(n) + h(n)$ โดยที่ : $g(n)$ = ค่าใช้จ่ายในการเคลื่อนที่จากโหนดหนึ่งไปยังอีกโหนดหนึ่ง สิ่งนี้จะแตกต่างกันไปในแต่ละโหนด $h(n)$ = การประมาณค่าของโหนดแบบฮิวริสติก นี้ไม่ใช่มูลค่าที่แท้จริง แต่เป็นต้นทุนโดยประมาณ

2.5.3.3 การทำงานของ A*



รูป 2.3 การทำงานของ A*

ตัวอย่างการทำงานของอัลกอริทึม A* โดยที่โหนดคือเมืองที่เชื่อมต่อกับถนน และ $h(x)$ คือระยะทางเป็นเส้นตรงไปยังจุดเป้าหมายดังรูป 2.3 โดยกำหนดแต่ละสีดังนี้

สีเขียว: เริ่มต้น , สีน้ำเงิน: เป้าหมาย , สีส้ม: จำนวนความต่างของระยะทาง โดยจากรูป จะเห็นได้ว่าเส้นทางเริ่มสีเขียวของ $h(a) = 4$ กับ $h(b) = 2$ เพื่อไปหาสีฟ้า นั้นมีระยะทางที่มากกว่าเพราะต้องไปทาง $h(c) = 4$ ต่อ ในขณะที่อีกเส้นทางเริ่มสีเขียวของ $h(d) = 4.5$ กับ $h(e) = 2$ ก็สามารถถึงสีฟ้าได้แล้วคือจุดต่อไป ดังนั้นเส้นทางที่สั้นที่สุดของภาพนี้จึงเป็นทางเริ่มสีเขียวของ $h(d) = 4.5$ กับ $h(e) = 2$

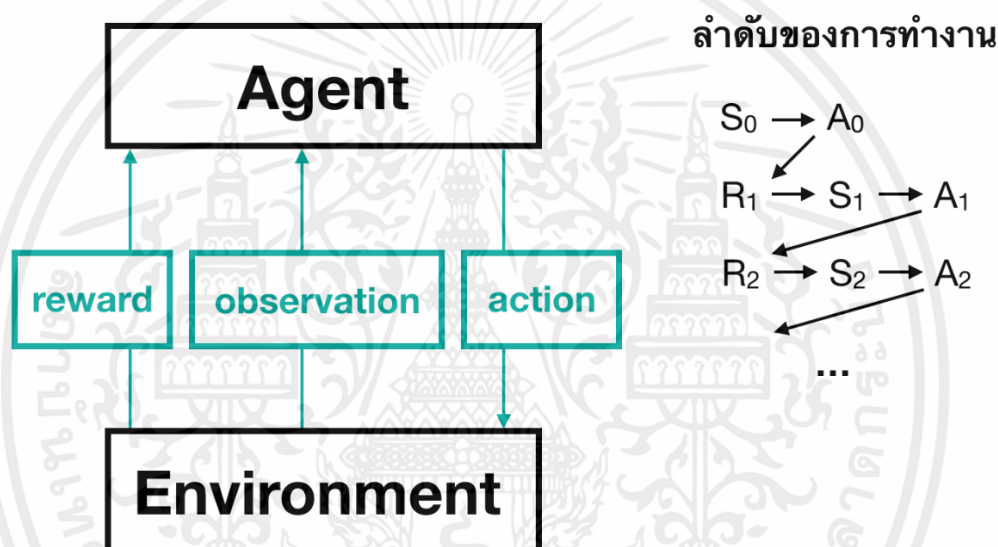
2.6 Reinforcement learning (RL)

Reinforcement learning เป็นวิธีการเรียนรู้แบบหนึ่งที่เกิดมาจากการปฏิสัมพันธ์ (Interaction) ระหว่างผู้เรียนรู้ (agent) กับสิ่งแวดล้อม (environment) ตัวอย่างของ Reinforcement learning ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

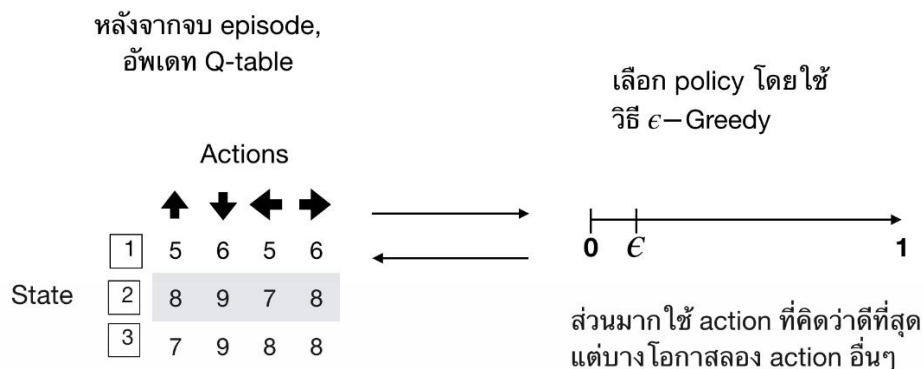
1. การเล่นเกม RPG ถ้าเราเป็นผู้เล่นที่ต้องการจะจบเกมนั้นให้ได้ วิธีการก็คือการเอาชนะศัตรูทั้งหมดภายในเกมและเรียนรู้ทักษะต่างๆ ที่สามารถใช้ได้เวลาต่อสู้โจมตีศัตรูให้ตาย
2. การเล่นเกม Player battle ground ถ้าเราเป็นผู้เล่น ก็ต้องสังเกตว่ามีศัตรูอยู่ใกล้ๆ หรือไม่ ถ้ามีศัตรูอยู่ใกล้เราต้องหันไปทางศัตรูและกดยิงเพื่อให้เราได้แต้มเยอะกว่าศัตรู

ใน Reinforcement learning การเรียนรู้เริ่มต้นที่จะสำรวจอยู่ที่ state ใดแล้ว S_0 ตัดสินใจทำบางอย่าง A_0 และได้ผลลัพธ์กลับมา R_1 เป็นแบบนี้ไปเรื่อย ๆ ลำดับของการทำงาน ดังรูป 2.4



รูป 2.4 ลำดับการทำงานของ Reinforcement learning (RL)

ในแผนผังนี้เราเริ่มต้นโดยการกำหนดผู้เรียนรู้ (agent) โดย agent สามารถสำรวจได้ว่าอยู่ที่ไหนของ environment (state) จากนั้น agent จะใช้การกระทำบางอย่าง (action) เช่น โจมตีหลังจากที่ใช้ action ไปแล้ว agent ก็จะได้ reward กลับมา จะใช้ Q-table ในการจำลอง agent ขึ้นมา และให้ agent เดินไปจนถึงจุดสิ้นสุด และอัปเดต Q-table ดังรูป 2.5



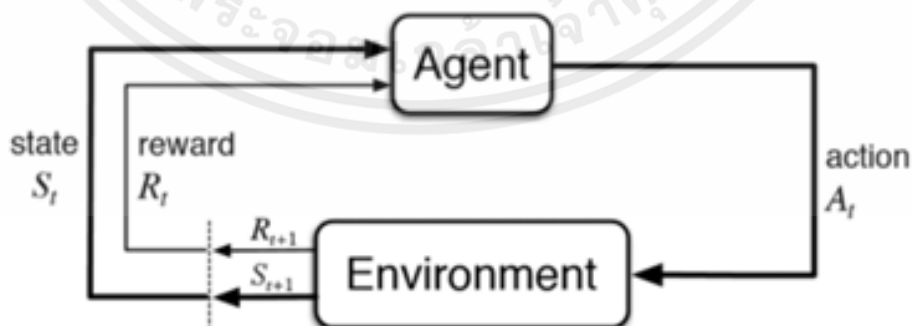
รูป 2.5 ตารางระหว่าง State-Action (Q-table)

การอัปเดต Q-table โดยใช้วิธี Monte Carlo และการเลือกชุดคำสั่งโดยใช้เทคนิค epsilon Greedy การอัปเดตเกิดขึ้นหลังจาก agent เดินไปจนจบ episode สมการของการอัปเดตเป็นดังนี้

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(G_t - Q(S_t, A_t)) \quad (2.2)$$

โดย G_t คือรางวัลหรือคะแนน (reward) ที่เราได้ตอนจบด่าน หลังจากออกใช้ action นั้นๆ ไป จะเห็นว่า ถ้าหลังจากใช้ action ไปแล้วเราได้คะแนนน้อยกว่า Q ใน state และ action นั้นๆ ก็จะลดลง แต่ถ้าเราได้คะแนนกลับมาเพิ่มขึ้นค่า Q ก็จะเพิ่มมากขึ้นดังสมการที่ 2.2

2.7 การฝึกบอสให้เก่งด้วย RL



รูป 2.6 Classic RL Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรเจกต์เกม Unity มีค่าต่อการออกแบบ GPU ตัดสินใจที่จะฝึกตัวแทนแค่ Boss ที่ชาญฉลาดซึ่งยากที่จะเอาชนะ โดยใช้เกมที่มีอยู่ในการทดลองการฝึกจะจำกัดจำนวนตัวละครที่จะฝึกได้สองแบบ: ผู้เล่นหนึ่งคน Player และ Boss เท่านั้น คือ Agent นอกจากนี้ เพื่อให้การฝึกง่ายขึ้น จะจำกัดฉากเกมให้อยู่ในสวนกล่องหรือ Environment อย่างง่าย เริ่มแรกจะฝึกตัวแทนผู้เล่น Player ของเรากับ Boss ที่อยู่กับทันที จากนั้นเราฝึก Boss ของเรากับตัวแทนผู้เล่นที่ได้รับการฝึกฝน จากนั้นตัวแทนผู้เล่นและ Boss จะผลัดกันฝึกซ้อมซ้ำๆ หลายรูปแบบ State ด้วยท่าทางการโจมตี Action ด้วยวิธีนี้ ในขณะที่ Boss สามารถกำจัดผู้เล่นได้ดีกว่า ผู้เล่นก็จะแข็งแกร่งขึ้น และ Boss จะต้องฉลาดขึ้นเพื่อเอาชนะ Player เป็นการได้รับ Reward ดังที่แสดงรูป 2.6

2.7.1 Proximal Policy Optimization (PPO)

การฝึก Boss ของเราให้กำจัด Player ที่หยุดนิ่ง และสลับกันได้โดยใช้ PPO (Proximal Policy Optimization) เช่น ในสภาพแวดล้อม environment นี้ จะฝึก Boss ของเราให้กำจัด Player ที่อยู่กับทันทีแบบสุ่ม แต่ละตอนของการฝึก จำกัด 250 ครั้ง ในแต่ละครั้ง จะได้เพิ่มรางวัลเป็น $(-1 / 250)$ ซึ่งหมายความว่าตัวแทนจะได้รับรางวัล -1 หาก Boss ของเราไม่สามารถเอาชนะ Player ได้ภายในเวลาที่กำหนด สิ่งนี้กระตุ้นให้ตัวแทนกำจัด Player โดยเร็วที่สุด หาก Player ถูกกำจัดได้สำเร็จ เราจะเพิ่มรางวัลให้ 1 ใน Unity จะสามารถใช้กับแมพหลายๆ อินสแตนซ์ multiple instances เพื่อให้การรวบรวมข้อมูลเป็นแบบคู่ขนาน parallelize และทำให้การฝึกเร็วขึ้น ใช้อัลกอริทึมดังสมการที่ 2.3

$$\begin{aligned}\nabla_{\theta} L &= \int S^{\mu(s)} \int A^{\nabla \theta} \pi_{\theta}(a|s) A(s, a) \\ &= E[\nabla_{\theta} \log \pi_{\theta}(a|s) A(s, a)]\end{aligned}\tag{2.3}$$

ในการเรียนรู้การเสริมแรง อัลกอริทึมที่อิงตามการไล่ระดับนโยบายให้กระบวนการที่สอดคล้องกันที่โดดเด่นสำหรับปัญหาพื้นที่ดำเนินการต่อเนื่อง จุดประสงค์ของอัลกอริทึมดังกล่าวทั้งหมดคือการเพิ่มผลตอบแทนที่คาดหวังไว้สูงสุด $L = E \sum_{t \in \tau} r_t$ การไล่ระดับที่ใช้บ่อยที่สุดของฟังก์ชันวัตถุประสงค์ L ที่มีเส้นฐานสามารถเขียนได้ในรูปแบบต่อไปนี้ โดยที่ S หมายถึงชุดของสถานะทั้งหมดและ A หมายถึงชุดของการกระทำทั้งหมดตามลำดับ $\mu(s)$ คือการกระจายตามนโยบายทั่วรัฐ π เป็นนโยบายสุ่มที่แมพสถานะ $s \in S$ กับการกระทำ $a \in A$ และ A เป็นฟังก์ชันที่ได้เปรียบ TRPO แทนที่ฟังก์ชันวัตถุประสงค์ L ด้วยวัตถุประสงค์ตัวแทน ซึ่งมีข้อจำกัดในขอบเขตของการอัปเดตจากนโยบายเก่าเป็นนโยบายใหม่ โดยเฉพาะโดยที่ $\pi_{\theta_{old}}$ เป็นนโยบายเก่าที่สร้างการดำเนินการ แต่พารามิเตอร์จะได้รับการแก้ไข

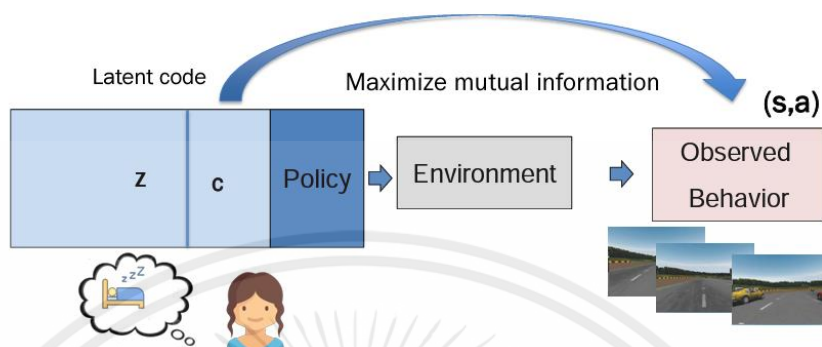
ในระหว่างกระบวนการฝึกอบรม $\pi\theta$ เป็นนโยบายที่เราพยายามปรับให้เหมาะสมแต่ไม่มากเกินไป ดังนั้นจึงเพิ่มขอบเขตบน δ เพื่อจำกัดความแตกต่างของ KL ระหว่าง $\pi\theta$ และ $\pi\theta_{old}$ PPO นั้นสามารถถือเป็น TRPO เวอร์ชัน โดยประมาณแต่ง่ายกว่ามาก มันตัดอัตราส่วนระหว่าง $\pi\theta_{old}$ และ $\pi\theta$ คร่าวๆ แล้วเปลี่ยนฟังก์ชันวัตถุประสงค์ตัวแทนให้อยู่ในรูปแบบสมการที่ 2.4

$$yL_{\theta CLIP} = E[\min(r(\theta)A, \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon)A)] \quad (2.4)$$

2.7.2 Generative Adversarial Imitation Learning (GAIL)

อัลกอริทึม แบบที่ 2 (GAIL) จะเพิ่มประสิทธิภาพการเรียนรู้ให้กับ Boss เพื่อเอาชนะ Player หรือผู้เล่นที่มีทักษะ โดยใช้ GAIL (Generative Adversarial Imitation Learning) เพื่อ Boss เหมือนมนุษย์มากขึ้นจากการหลบหลีก Player หรือผู้เล่นได้จากรางวัลโจมตีมาของผู้เล่นทำให้เกมน่าตื่นเต้นมากขึ้น เราสามารถใช้ Generative Adversarial Imitation Learning (GAIL) เครื่องมือจาก ML Agent ของ Unity ช่วยให้เราสามารถเล่นเกมและบันทึกพฤติกรรมของ Boss ในไฟล์สาธิตหรือในการ Training ได้ ใน GAIL โครงข่ายประสาทเทียมที่สองเรียนรู้ที่จะแยกแยะระหว่างสถานะ/การดำเนินการของไฟล์สาธิต และตัวแทน โดยจะสร้างรางวัล (Reward) เพื่อที่จะวัดความสามารถว่าเอาชนะผู้เล่นได้จริงตามที่หวัง สามารถ "หลอก" หรือ หลบหลีก Player ได้ระหว่างที่เล่น โดยพื้นฐานแล้ว GAIL จะค่อยๆ ส่งผลให้ ตัวแทนหรือ Boss เลียนแบบการกระทำของ Player หรือผู้เล่น ได้ดีขึ้นด้วยวิธีการของ GAIL การกำหนด ปัญหาการนำทางในสภาพแวดล้อมหรือ Environment เป็นกระบวนการตัดสินใจของ Markov (MDP) สถานะ s_t ประกอบด้วยภาพความลึก x_t และแรงไปยังเป้าหมายที่ต้องการ F^{\rightarrow} idesired แต่ละการกระทำ ที่สอดคล้องกับคำสั่งที่เคลื่อนไหว u_t ที่จะดำเนินการโดยหุ่นที่เคลื่อนที่ เครือข่ายที่มีการกำหนด พารามิเตอร์สองเครือข่าย เครือข่ายตัวสร้าง (นโยบาย) $\pi\theta$ และเครือข่ายผู้เลือกปฏิบัติ $D\omega$ ได้รับการอัปเดตในลักษณะที่สลับกันในช่วงตอนการฝึก หรือ Training ดังรูป 2.7

$$L_I(\pi_\theta, Q_\psi) = \mathbb{E}_{c \sim p(c), a \sim \pi_\theta(\cdot|s,c)} [\log Q_\psi(c|s,a)] + H(c) \\ \leq I(c; s, a)$$



รูป 2.7 อัลกอริทึม Info GAIL

2.8 AI

AI ย่อมาจากคำว่า Artificial Intelligence คือเทคโนโลยีที่จำลองความฉลาดของมนุษย์ โดยการพัฒนาระบบอัจฉริยะที่มีความสามารถในการรับรู้ เรียนรู้ ใช้เหตุผล และตัดสินใจเลือก ทางเลือกที่ดีที่สุด จากการวิเคราะห์ข้อมูลที่เกี่ยวข้อง พิจารณาทางเลือกต่าง ๆ และผลลัพธ์ของทางเลือกนั้น ๆ ที่เป็นไปได้ทั้งหมด ภายใต้สภาวะแวดล้อมหรือ เงื่อนไขที่กำหนด

2.8.1 การจัดการอินพุต และการรับรู้ (Input handler and perception)

การรับรู้ของปัญญาประดิษฐ์คือสิ่งของในสภาวะแวดล้อมที่เราต้องการให้ตัวแทนในเกมส์ สมองตอบ ซึ่งทำได้โดยการใช้ระบบการรับรู้ซึ่งแบ่งเป็น

1. ชนิดของการรับรู้ (Perception type) ชนิดของอินพุตมีหลายชนิดรวมทั้ง Boolean integer floating point และอื่น ๆ ซึ่งอาจรวมถึงการรับรู้สถิตติ (Static perception) เช่น การรับรู้ที่ต้องการสำหรับลอจิก
2. การปรับสม่ำเสมอ (Update regularity) การรับรู้ต่างชนิดกันอาจต้องการปรับตามความจำเป็น เพราะปกติการรับรู้จะไม่ปรับบ่อย
3. เวลาในการตอบสนอง (Reaction time) การใช้เวลาในการตอบสนองของตัวแทนจะทำให้ในตัวเกมมีลักษณะคล้ายมนุษย์มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เส้นแบ่ง (threshold) คือค่าที่น้อยที่สุดและมากที่สุดที่ปัญญาประดิษฐ์จะตอบสนอง และค่านี้จะเปลี่ยนแปลงตามเหตุการณ์ในเกม
5. ความสมดุลของโหลด (Load balancing) การทำความสมดุลของโหลดเพื่อไม่ให้ระบบการรับรู้ใช้ resource มากเกินไป

2.8.2 การเดินทาง (Navigation)

การเดินทางจากจุด A ไปยังจุด B เกี่ยวข้องกับสภาวะแวดล้อมที่ใหญ่และมีความซับซ้อน ซึ่งมีความหลากหลายลักษณะภูมิประเทศ สิ่งกีดขวาง วัตถุเคลื่อนที่ และอื่น ๆ การเดินทางมีงานหลัก ๆ อยู่ 2 อย่าง คือ การหาเส้นทาง (pathfinding) และการหลบหลีกสิ่งกีดขวาง (obstacle avoidance) วิธีการที่ทำให้ตัวแทนเดินทางจากจุด A ไปยัง B มีหลายวิธี เช่น Cell decomposition, Navigation Graph เป็นต้น

2.8.3 การตัดสินใจ (Decision making) และการอนุมาน (Inference)

ระบบการตัดสินใจเป็น โครงสร้างพื้นฐานที่จะบ่งบอกถึงชนิดของเครื่องประมวลผล ปัญญาประดิษฐ์ที่ต้องการสร้าง และการอนุมานเป็นการสรุปแบบมีลอจิกและมีเหตุผลโดยพิจารณาจากความจริงหรือสาเหตุที่เป็นจริง ดังนั้นในเกมส์ ศัตรูที่ถูกควบคุมโดยปัญญาประดิษฐ์จะรับข้อมูลเกี่ยวกับโลก และพยายามตัดสินใจอย่างฉลาดและมีเหตุผลในการตอบสนอง ความแตกต่างระหว่างระบบการตัดสินใจจะบ่งบอกว่าผู้สร้างเกมจะเลือกใช้ระบบชนิดใดหรือจะใช้หลายระบบรวมกัน โดยความแตกต่างของระบบแยกออกได้เป็น

1. ชนิดของคำตอบ (Type of solutions) เช่นเป็นเกมชนิดยุทธศาสตร์ (Strategic) หรือ กลวิธี (tactical) ถ้าเป็นยุทธศาสตร์เกม ต้องการคำตอบที่เป็นระยะยาว มีจุดมุ่งหมายสูงซึ่งมักจะเกี่ยวข้องกับการมีการกระทำหลายที่ต้องทำ ส่วนคำตอบของกลวิธีเกมจะเป็นระยะสั้น จะมีจุดมุ่งหมายต่ำ และโดยปกติจะเกี่ยวข้องกับการกระทำทางกายภาพ (physical act) หรือความสามารถ
2. ปฏิกริยาของตัวแทน (Agent reactivity) ความต้องการให้ตัวแทนในเกมมีปฏิกริยามากน้อยเพียงใด
3. ความเป็นจริงของระบบ (System Realism) ตัวละครแต่ละตัวในเกมจะต้องทำในสิ่งที่ควรจะทำอย่างฉลาด และควรมีความเป็นมนุษย์อยู่ด้วย นั่นคือตัวแทนจำเป็นที่ต้องแสดงถึงความอ่อนแอของมนุษย์
4. ประเภท (Genre) เกมที่ต่างประเภทกันต้องการปัญญาประดิษฐ์ที่ไม่เหมือนกัน

5. เนื้อหา (Content) ในเนื้อหาของเกมบางครั้งจะมีส่วนที่เป็นการเล่นที่เป็นเคสพิเศษ เช่น Black & White เกมต้องการปัญญาประดิษฐ์พิเศษ สำหรับการสอนสัตว์ให้มีพฤติกรรม โดยการแสดงให้ดูหรือทำในสิ่งที่ถูกใจให้ดู
6. แพลตฟอร์ม (Platform) ถึงแม้ว่าการพิจารณาในเรื่องนี้ไม่ค่อยสำคัญนักแต่ในบางครั้งก็ยังคงพิจารณาว่าเกมที่สร้างจะถูกนำไปเล่นที่ใด
7. ข้อจำกัดทางด้านความบันเทิง (Entertainment limitations) การสร้างความสมดุลระหว่างความยากของเกมและความสมจริง เป็นสิ่งที่เป็นขั้นตอนสุดท้ายของความสำเร็จของเกม นั่นคือควรจะทำในสิ่งที่มนุษย์ทำได้ และไม่ควรจะทำให้สิ่งที่มนุษย์ไม่สามารถทำได้ และระดับความยากของเกมเป็นอีกสิ่งหนึ่งที่ปัญญาประดิษฐ์เข้ามาช่วยได้ ทั้งนี้รวมทั้งการทำให้ความตั้งใจของตัวแทนส่งต่อไปที่ผู้เล่น

บทที่ 3

การออกแบบ

ในการดำเนินโครงการ การสร้างและออกแบบเกม “3D RPG GAME DEVELOPMENT” มีขั้นตอนการวิเคราะห์และออกแบบระบบ ดังนี้

3.1 Requirement

ตาราง 3.1 Requirement

ID	Detail	Type	Priority
1	เกมสามารถแสดงผลเป็นภาพ 3 มิติได้	-Player -Gameplay -Functional	Must Have
2	เกมมีมุมมองการแสดงผลแบบบังคับทิศทางมองเห็นได้	-Player -Gameplay -Functional	Must Have
3	เกมมีระบบการควบคุมการเคลื่อนไหว เดินหน้า/ซ้าย/ขวา/หลัง และการกระโดด ทิศทางแบบ 3 มิติได้	-Player -Gameplay -Functional	Must Have
4	เกมมีระบบการเล็งหรือล็อกเป้าเพื่อโจมตีศัตรูได้แม่นยำขึ้น	-Player -Gameplay -Functional	Must Have
5	เกมมีระบบเพิ่มเลเวลตัวละครได้	-Player -Gameplay -Functional	Must Have
6	เกมมีระบบ Exp เพื่อแสดงให้เห็นว่าผู้เล่นใกล้จะไป Level ถัดไปตอนไหน	-Player -Gameplay -Functional	Must Have

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	เกมมีการแยกโหมดให้เลือกเล่นตามทักษะของผู้เล่น	-Player -Gameplay -Functional	Must Have
8	เกมมีระบบการวิ่งเร็วขึ้นได้	-Player -Gameplay -Functional	Must Have
9	เกมมีระบบมินิแมพเพื่อให้เห็นศัตรูในระยะที่ไกลที่สุดได้	-Player -Gameplay -Functional	Must Have
10	เกมมีระบบ Score ของผู้เล่นเมื่อทำการฆ่าศัตรูได้	-Player -Gameplay -Functional	Must Have
11	เกมมีระบบ HP ของผู้เล่นเพื่อแสดงว่า HP ของผู้เล่นมีเท่าไร	-Player -Gameplay -Functional	Must Have
12	เกมมีระบบลด HP ของผู้เล่นเมื่อถูกศัตรูโจมตี	-Player -Gameplay -Functional	Must Have
13	เกมมีระบบแสดง Level ของผู้เล่น	-Player -Gameplay -Functional	Must Have
14	ผู้เล่นสามารถโจมตีระยะไกลได้โดยใช้อาวุธ	-Player -User Interface -Functional	Must Have
15	เกมมีระบบการเกิดของศัตรู	-Player -User Interface -Functional	Must Have
16	เกมมีประเภทของศัตรูภายในเกม 3 ประเภท	-Enemy -Functional	Must Have

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17	เกมมีศัตรูที่เป็น Boss ประจำแต่ละด่าน	-Enemy -Functional	Must Have
18	ศัตรูสามารถเข้าไปหาตำแหน่งที่อยู่ของตัวละครได้	-Enemy -Functional	Must Have
19	ศัตรูสามารถโจมตีตัวละครได้	-Enemy -Functional	Must Have
20	Boss สามารถเรียนรู้เอาชนะ Player ด้วยตนเองได้	-Enemy -Functional	Must Have
21	เกมมีหน้า Lobby ให้ผู้เล่นเตรียมตัวก่อนเริ่มเกม	-Gameplay -Map -Functional	Must Have
22	ตัวละครสามารถชนะเกมได้ ด้วยการจัดการ Boss ภายในเกม	-Gameplay -Functional	Must Have
23	ศัตรูสามารถตายได้	-Enemy -Gameplay -Functional	Must Have
24	Player สามารถ Warp ไปห้อง Boss ได้	-Player -Gameplay -Functional	Must Have

3.2 การออกแบบในส่วนของเกม

3.2.1 Game Concept

ตัวเกมนั้นสร้างมาเพื่อให้ความสนุก และคิดวิธีที่จะสามารถเอาชนะศัตรูได้ ในการเล่นแต่ละครั้ง มีระบบการเพิ่ม Level ของตัวละครในการฆ่าศัตรูได้และจะเพิ่มความยากของการเพิ่ม Level ในทุกๆ การเพิ่ม Level บวก 2000 Max Exp ใน Level ที่เพิ่งเพิ่มได้ ประกอบกับการมองตัวละครของผู้เล่นว่าหันมุมมองถูกทางหรือไม่ เพราะทิศทางมีผลกับการโจมตีแบบจริงๆ โดยเกมจะมีมุมมองแบบ 3D และทำเป็นแบบ Single Player คือสามารถเล่นได้คนเดียวเพื่อหลีกเลี่ยงการแข่งขันกับผู้เล่นอื่นให้ไม่

รู้สึกกดดันในการเล่นแต่เพิ่มแรงกระตุ้นคือความรู้สึกดี สนุกสนาน เมื่อได้ Level ตัวละครเพิ่มขึ้นจากการฆ่าศัตรู หรือเอาชนะได้

3.2.2 Story และ Setting

ตัว Setting เนื้อเรื่องภายในเกมนั้นจะเป็นแนวย้อนยุค และผสมผสานกับมอนสเตอร์ หรือ หมายถึง สัตว์ประหลาด สิ่งมีชีวิตประหลาดที่ปรากฏตามวรรณกรรม นำมาใช้กับเกมในส่วนของตัวศัตรูที่ใช้โดยมี Boss ที่เป็นหัวหน้าคอยสั่งการและดูแลและพื้นที่นั้น ซึ่งมีสภาพแวดล้อมไม่เหมือนกันในแต่ละ โหมดเพื่อเพิ่มความยากให้กับผู้เล่น ซึ่งตัวของผู้เล่นจะสวมบทบาทเป็นนักฆนุหญิงที่เข้ามาจะพิชิต Boss เพื่อหวังจะเอาสมบัติกลับไป โดยจะต้องเอาชนะศัตรูทั้งหมดภายในแมพพร้อม Boss ให้ได้

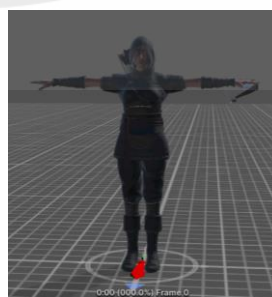
3.2.3 Graphic

ในตัวเกมนั้นจะมีมุมมองในรูปแบบ 3D แบบบังคับทิศทาง การเคลื่อนไหวและมุมมองที่สมจริงของมนุษย์ ลักษณะของภาพในเกมจะใช้ 3D โมเดลทั้งหมด ยกเว้น Icon ต่างๆ

3.2.4 Gameplay

3.2.4.1 Player

- 1) Player จะรับบทสวมบทบาทเป็นนักฆนุหญิง ลักษณะเหมือนมนุษย์จริงๆ ไม่สามารถปรับเปลี่ยนได้
- 2) Player สามารถเคลื่อนไหว เดินหน้า / ซ้าย / ขวา / หลัง รวดเร็วและ กระโดดได้
- 3) Player สามารถดูแผนที่หรือ มินิแมพได้เพื่อดูศัตรูที่อยู่ใกล้และกำลังจะ โจมตี
- 4) Player สามารถกดคลิกขวาเพื่อเล็งหรือเพิ่มความแม่นยำการ โจมตีได้
- 5) Player สามารถ Level Up ได้จากการเก็บ Exp โดยการฆ่าศัตรู



รูป 3.1 ตัวละครผู้เล่น Player หญิง(นักฆนุ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

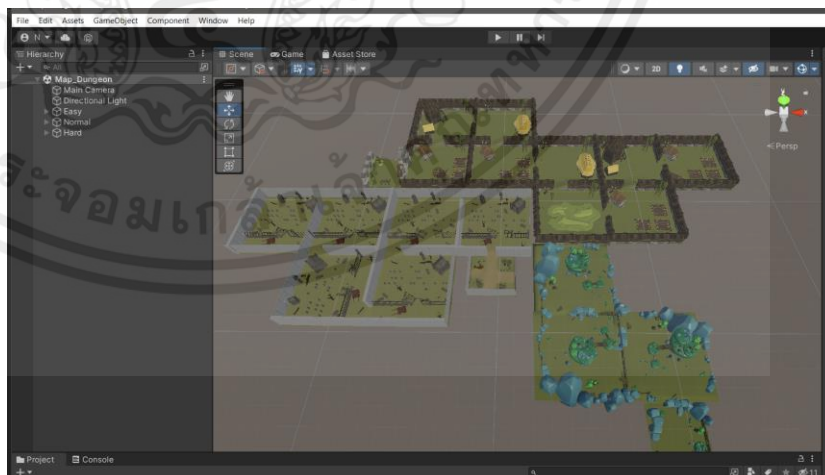
3.2.4.2 Progression

การ Level Up ของ Player ได้จากการเก็บ Exp โดยการฆ่าศัตรูและทุกๆ การเพิ่ม Level จะเพิ่มความยากในการไป Level ถัดไป บวก 2000 ต่อการ Up Level ดังนั้น Player จะต้องฆ่าศัตรูมากขึ้นเพื่อจะไป Level ต่อไป

3.2.4.3 Level Design

โหมด (Mode) : โหมดจะมีให้เลือกแบ่งเป็น 3 แบบคือ Easy Medium Hard โดยจะมีห้อง Boss หรือ Boss Room แยกพื้นที่เพื่อให้สู้กัน 1 ต่อ 1 กับ Player

- 1) โหมด Easy จะมีขนาด 300*300 ในพื้นที่การจัดการศัตรู พื้นดินระดับเดียวกัน มีทั้งต้นไม้ขนาดใหญ่และเล็ก สุ่ม Random ศัตรู 10-15 ทั้งหมดจะเป็น 2 ประเภทและมี Boss Easy 1 ประเภท
- 2) โหมด Medium จะมีขนาด 600*600 ในพื้นที่การจัดการศัตรู พื้นดินระดับเดียวกัน มีทั้งต้นไม้ขนาดใหญ่และเล็ก สุ่ม Random ศัตรู 10-20 ทั้งหมดจะเป็น 2 ประเภทและมี Boss Medium 1 ประเภท
- 3) โหมด Hard จะมีขนาด 1000*1000 ในพื้นที่การจัดการศัตรู พื้นดินระดับเดียวกัน มีทั้งต้นไม้ขนาดใหญ่และเล็ก สุ่ม Random ศัตรู 10-30 ทั้งหมดจะเป็น 2 ประเภทและมี Boss Hard 1 ประเภท



รูป 3.2 ออกแบบแมพทั้ง 3 โหมด ทั้ง Easy , Medium , Hard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4.4 Enemy

ศัตรูที่ออกแบบไว้จะแบ่งเป็น 3 ประเภท

- 1) ศัตรูที่มีลักษณะเป็นโครงกระดูก : ศัตรูประเภทนี้จะวิ่งช้าและโจมตีช้ากว่าประเภทอื่น แต่จะโจมตีแรง Player สามารถหนีการโจมตีได้ง่าย อาวุธถือดาบสั้น



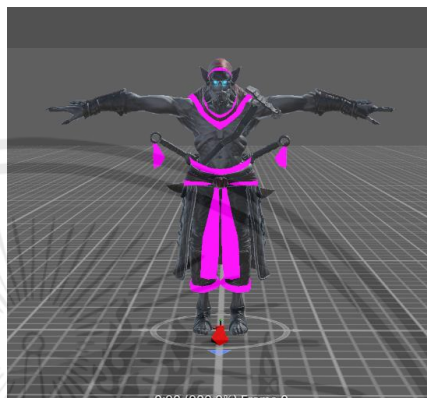
รูป 3.3 ตัวละครศัตรู หรือ Enemy Skeleton

- 2) ศัตรูที่มีลักษณะเหมือนหมาป่าเดิน 4 ขา : ศัตรูประเภทนี้จะวิ่งเร็วและโจมตีเร็วกว่าประเภทอื่น แต่จะโจมตีได้เบากว่ามีความรวดเร็วนิเมชันการโจมตีโดยใช้ปากกัด Player เพื่อทำคาเมท สามารถหนีการโจมตีได้ยาก



รูป 3.4 ตัวละครศัตรู หรือ Enemy NightmareDragon

- 3) ศัตรูที่มีลักษณะมีฟ้าคลุมสวมและเป็นหน้าปีศาจ : ศัตรูประเภทนี้จะวิ่งเร็วและโจมตีเร็วกว่าประเภทอื่น สามารถทำคาเมทได้รุนแรงประกอบกับมีอนิเมชันการโจมตีที่รวดเร็ว แม่นยำ ทำให้ Player สามารถเอาชนะได้ยากขึ้น

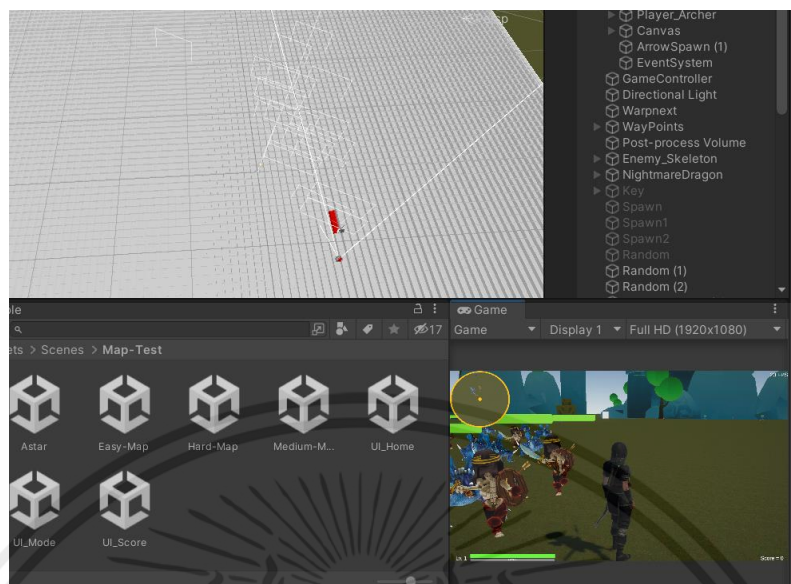


รูป 3.5 ตัวละคร Boss

3.2.4.5 AI Pathfinder

AI ที่นำมาใช้คือ สาขาหนึ่งของวิทยาศาสตร์คอมพิวเตอร์ที่เกี่ยวข้องกับการพัฒนาระบบที่สามารถจำลองหรือคิดแทนการทำงานของมนุษย์โดยมีการเทคโนโลยีคอมพิวเตอร์ต่าง ๆ เช่นการเรียนรู้เชิงลึก (Deep Learning), การประมวลผลภาษาธรรมชาติ (Natural Language Processing), การค้นหาเส้นทาง (Pathfinding) โดยใช้ Shortest Path Algorithms เป็นอัลกอริทึมที่ใช้ในการค้นหาเส้นทางที่สั้นที่สุด หรือ A Star (A*) และการวิเคราะห์ข้อมูล (Data Analytics) เป็นต้น

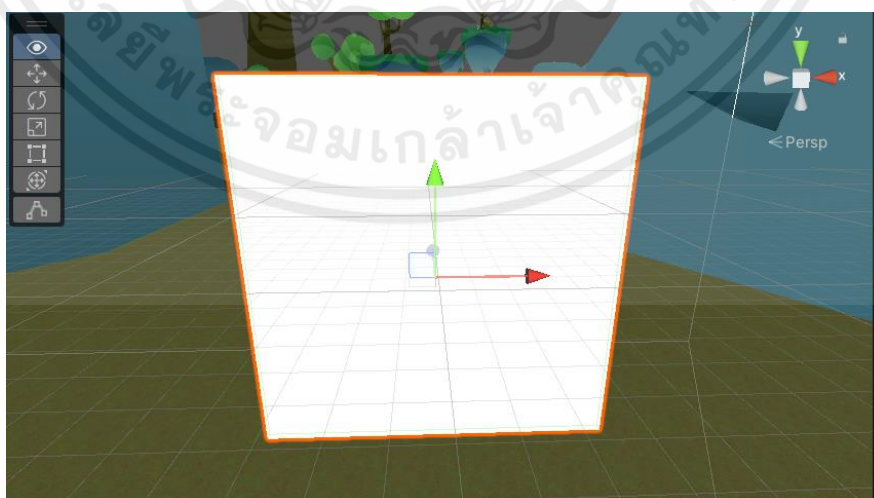
โดยหลักการ A Star ที่นำมาใช้กับตัวเกมคือศัตรู 2 ประเภท คือ ศัตรูที่มีลักษณะเป็นโครงกระดูก และ ศัตรูที่มีลักษณะเหมือนหมาป่าเดิน 4 ขา ซึ่งจะช่วยให้ Object หรือตัวของศัตรูไปยังจุดหมายได้อย่างรวดเร็วและถูกต้อง โดยจะระบุ ตัวของ Player และ ศัตรูเป็น Node จะเช็คสิ่งกีดขวาง โดยตัวเกมจะสร้าง Graph ระบุเส้นสีแดงคือเส้นทางที่สั้นที่สุดของ Enemy เพื่อไปหา Player ดังภาพ



รูป 3.6 หาเส้นทางที่สั้นที่สุดโดยการสร้าง Graph ขึ้นเช็คตำแหน่งของ Node

3.2.4.6 ประตูวาร์ป (Warp) เพื่อไปห้อง Boss

ในการเปลี่ยน Scene ในแมพที่สร้างเดิมอยู่แล้ว หรือ การที่เราสร้าง Scene ใหม่แล้วต้องการเปลี่ยนไปจุดนั้นจะใช้ประตูวาร์ป เพื่อให้สามารถไปจุดนั้นได้ โดยการสร้าง Object ขึ้นมาเป็น cube หรือรูปทรงกล่องแล้วปรับขนาดใหม่



รูป 3.7 วาร์ป (Warp)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4.7 Objective

การชนะของเกมนี้จะเป็นการที่ผู้เล่น ต่อสู้และชนะบอสของแต่ละโหมดที่เลือกเล่นทั้ง Easy , Medium , Hard และผู้เล่นจะแพ้เมื่อ HP ของตัวละครน้อยกว่าหรือเท่ากับ 0

3.2.4.8 ตัวควบคุมระบบเกม (Level Manager)

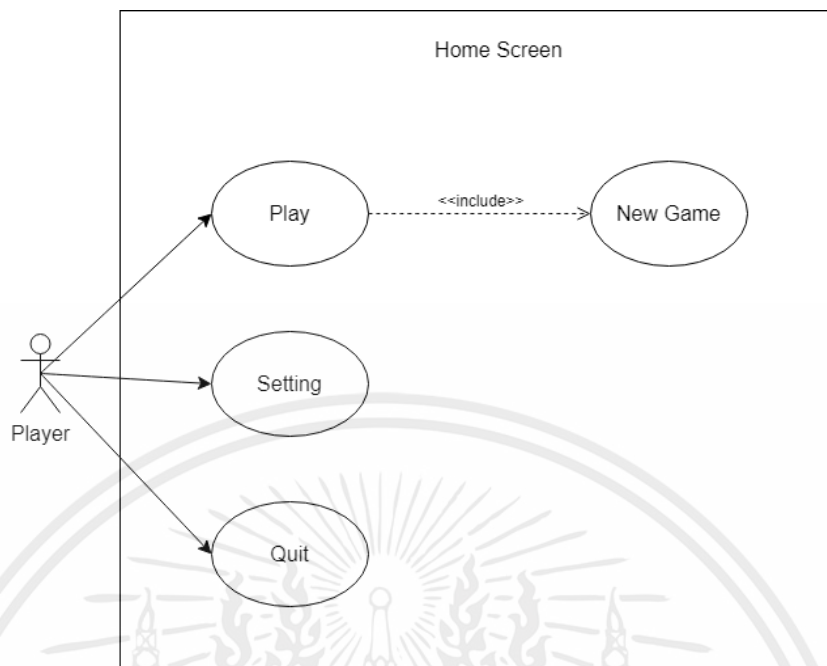
Level Manager จะรวมฟังก์ชันที่ใช้ของแต่ละ Object ที่สร้างขึ้นมา เพื่อให้ง่ายต่อการแก้ไขพารามิเตอร์ต่างๆ ของ Game Engine ซึ่งเป็นไฟล์โดยเฉพาะของ Unity ที่ไว้สร้างใช้ควบคุมระบบเกม โดยระบบที่นำเข้าไปใช้ในไฟล์นี้คือ Level ของตัวละคร การได้ Exp จากการฆ่าศัตรู แล้ว HP ของศัตรูเหลือ 0 แสดง Animation การตายของตัวละครก็จะได้รับ Level เพิ่มขึ้น และได้รับ Score บวก 1 เพิ่มแสดงค่าบนตัวเกม รวมถึงเปลี่ยนค่า Max Exp ที่ต้องเก็บให้ครบถึงจะไป Level ถัดไปจาก 0/2000 เป็น 0/4000 ทุกๆ ครั้งที่มีการเพิ่ม Level

3.2.4.9 Machine Learning (RL)

Boss จะมีการใช้ Machine Learning เพื่อให้เรียนรู้การเอาชนะ Player ได้ด้วยตัวเอง โดยการ Training Boss ด้วย Python API จากเงื่อนไขการเอาชนะด้วยการโจมตี Player ก็จะได้รับรางวัลหรือ Reward เพื่อกระตุ้นการเอาชนะจากการใช้ Reinforcement Learning และคู่มือผลลัพธ์

3.3 แผนภาพ Use Case Diagram

ตัวเกมจะมีหน้าจอ Scene หลักๆ 4 หน้าจอ ได้แก่ Home , Mode , Game Map (Mode Easy , Medium , Hard) , Score โดยในแต่ละหน้าจะมีหน้า UI และระบบของเกมต่างๆ ให้ผู้เล่นสามารถใช้ได้ภายในเกม ดังรูป



รูป 3.8 Use Case ของหน้าจอ Home Screen

โดยส่วนรายละเอียดของ Use Case ของหน้าจอ Home Screen จะแสดงดังตาราง 3.2 ถึง 3.5

Use Case Detailed Semantics Home Screen

ตาราง 3.2 การเริ่มเกม (Play) Use Case

Extension Use Case : เริ่มเกม (Play)
Use case ID : 1
Actors : ผู้เล่น (Player)
Goal : เข้าสู่หน้าเริ่มเกม
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่ตัวเลือกการเริ่มเกม
Trigger : กดปุ่ม Play

ตาราง 3.3 การเริ่มเกมใหม่ (New Game) Use Case

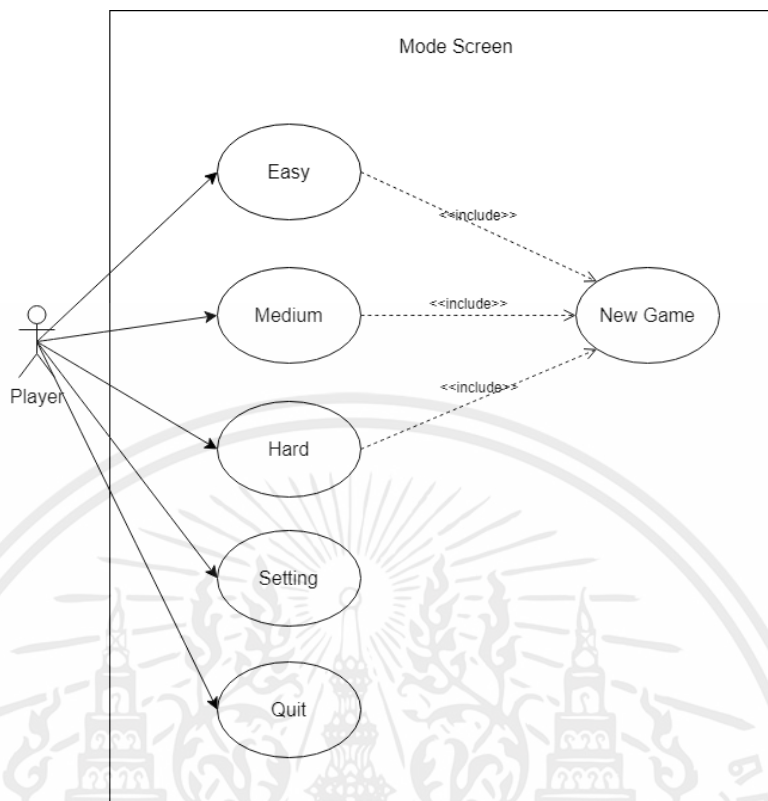
Extension Use Case : เริ่มเกมใหม่ (New Game)
Use case ID : 2
Actors : ผู้เล่น (Player)
Goal : เข้าสู่หน้าเริ่มเกมใหม่
Preconditions : ผู้ใช้กดปุ่ม Start Game
Postconditions : เข้าสู่หน้าเริ่มเกมใหม่ และเข้าสู่ Mode ที่ให้เลือก
Trigger : กดปุ่ม Play

ตาราง 3.4 การเข้าหน้าตั้งค่า (Setting) Use Case

Extension Use Case : เข้าหน้าตั้งค่า (Setting)
Use case ID : 3
Actors : ผู้เล่น (Player)
Goal : เข้าสู่หน้าตั้งค่า
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่หน้าจอการตั้งค่า
Trigger : กดปุ่ม Setting

ตาราง 3.5 การออกจากเกม (Quit) Use Case

Extension Use Case : ออกจากเกม (Quit)
Use case ID : 18
Actors : ผู้เล่น (Player)
Goal : ออกจากเกม
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : ออกจากเกมและปิด Client เกม
Trigger : กดปุ่ม Quit หรือ ESC



รูป 3.9 Use Case ของหน้าจอ Mode Screen

โดยส่วนรายละเอียดของ Use Case ของหน้าจอ Mode Screen จะแสดงดังตาราง 3.6 ถึง 3.11

Use Case Detailed Semantics Mode Screen

ตาราง 3.6 การ Mode (Easy) Use Case

Extension Use Case : Mode (Easy)
Use case ID : 5
Actors : ผู้เล่น (Player)
Goal : เข้าสู่ Mode (Easy)
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่หน้าจอ Mode (Easy)
Trigger : กดปุ่ม Easy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.7 การ Mode (Medium) Use Case

Extension Use Case : Mode (Medium)
Use case ID : 6
Actors : ผู้เล่น (Player)
Goal : เข้าสู่ Mode (Medium)
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่หน้าจอ Mode (Medium)
Trigger : กดปุ่ม Medium

ตาราง 3.8 การ Mode (Hard) Use Case

Extension Use Case : Mode (Hard)
Use case ID : 7
Actors : ผู้เล่น (Player)
Goal : เข้าสู่ Mode (Hard)
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่หน้าจอ Mode (Hard)
Trigger : กดปุ่ม Hard

ตาราง 3.9 การเริ่มเกมใหม่ (New Game) Use Case

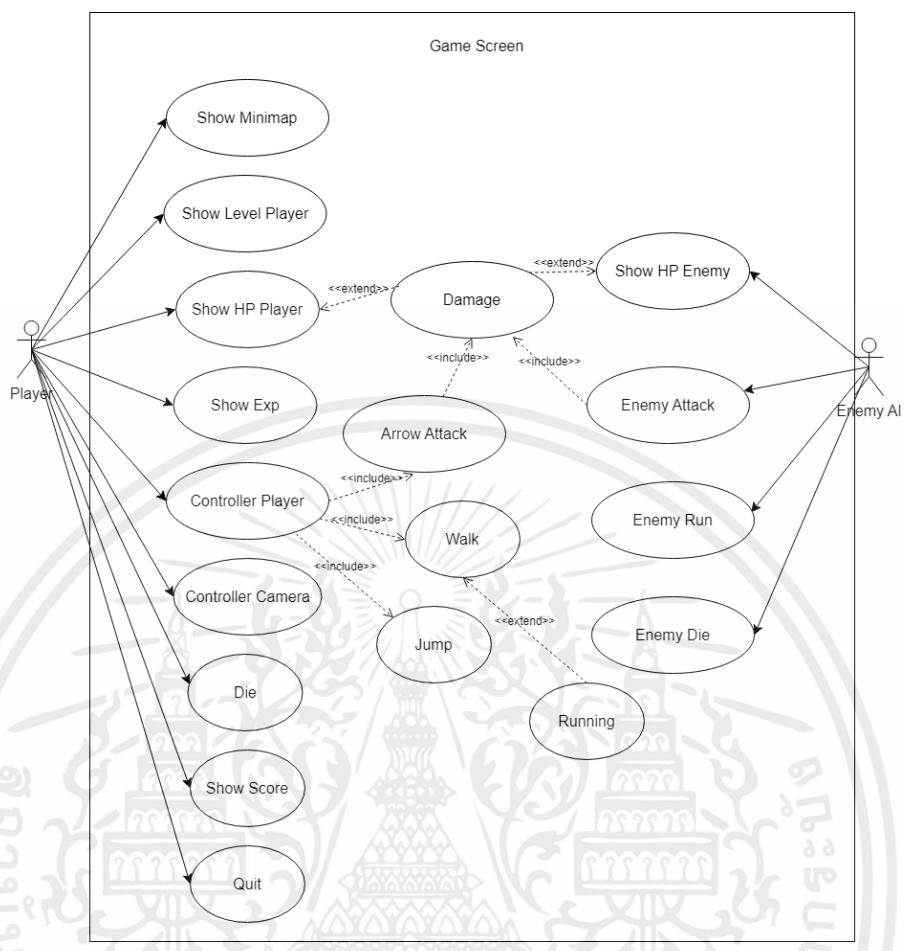
Extension Use Case : เริ่มเกมใหม่ (New Game)
Use case ID : 8
Actors : ผู้เล่น (Player)
Goal : เข้าสู่หน้าเริ่มเกมใหม่
Preconditions : ผู้ใช้กดปุ่ม Start Game
Postconditions : เข้าสู่หน้าเริ่มเกมใหม่ และเข้าสู่ Mode ที่ให้เลือก
Trigger : กดปุ่ม Easy Medium หรือ Hard

ตาราง 3.10 การเข้าหน้าตั้งค่า (Setting) Use Case

Extension Use Case : เข้าหน้าตั้งค่า (Setting)
Use case ID : 9
Actors : ผู้เล่น (Player)
Goal : เข้าสู่หน้าตั้งค่า
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่หน้าจอการตั้งค่า
Trigger : กดปุ่ม Setting

ตาราง 3.11 การออกจากเกม (Quit) Use Case

Extension Use Case : ออกจากเกม (Quit)
Use case ID : 18
Actors : ผู้เล่น (Player)
Goal : ออกจากเกม
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : ออกจากเกมและปิด Client เกม
Trigger : กดปุ่ม Quit หรือ ESC



รูป 3.10 Use Case ของหน้าจอ Game Screen

โดยส่วนรายละเอียดของ Use Case ของหน้าจอ Game Screen จะแสดงดังตาราง 3.12 ถึง 3.23

Use Case Detailed Semantics Game Screen

ตาราง 3.12 Show Minimap Use Case

Extension Use Case : Show Minimap
Use case ID : 11
Actors : ผู้เล่น (Player)
Goal : มี Minimap แสดงบนหน้าจอเพื่อคู่มือ
Preconditions : ผู้ใช้สามารถใช้งานได้เลย
Postconditions : แสดงบน UI แบบวงกลม
Trigger : -

ตาราง 3.13 Show Level Player Use Case

Extension Use Case : Show Level Player
Use case ID : 12
Actors : ผู้เล่น (Player)
Goal : มี Level ของ Player แสดงบนหน้าจอ
Preconditions : ผู้ใช้สามารถใช้งานได้เลย
Postconditions : แสดงบน UI แบบ ตัวเลข
Trigger : -

ตาราง 3.14 Show HP Player Use Case

Extension Use Case : Show HP Player
Use case ID : 13
Actors : ผู้เล่น (Player)
Goal : มี HP Bar ของ Player แสดงบนหน้าจอ
Preconditions : ผู้ใช้สามารถใช้งานได้เลย
Postconditions : แสดงบน UI แบบสี่เหลี่ยมผืนผ้า เขียว
Trigger : -

ตาราง 3.15 Show EXP Use Case

Extension Use Case : Show EXP
Use case ID : 14
Actors : ผู้เล่น (Player)
Goal : มี EXP Bar ของ Player แสดงบนหน้าจอ
Preconditions : ผู้ใช้สามารถใช้งานได้เลย
Postconditions : แสดงบน UI แบบสี่เหลี่ยมผืนผ้าด้านล่างต่อจาก HP Bar สีเหลือง
Trigger : -

ตาราง 3.16 Controller Player Use Case

Extension Use Case : Controller Player
Use case ID : 15
Actors : ผู้เล่น (Player)
Goal : ใช้งานฟังก์ชันใดก็จะแสดง Action นั้น
Preconditions : ผู้ใช้สามารถบังคับตัวละครได้
Postconditions : แสดง Action ของตัวละคร ตามที่ตั้งค่า Animator ไว้
Trigger : กดปุ่ม W (เดินหน้า) , A (เดินซ้าย) , S (เดินขวา) , D (เดินขวา) , คลิกขวาเลง , คลิกซ้ายยิงโจมตี , Space Bar (กระโดด)

ตาราง 3.17 Die Use Case

Extension Use Case : Die
Use case ID : 16
Actors : ผู้เล่น (Player)
Goal : ตายเมื่อ HP น้อยกว่าหรือเท่ากับ 0
Preconditions : ผู้ใช้สามารถแสดง Action ตายได้
Postconditions : แสดง Action ของตัวละคร ตามที่ตั้งค่า Animator ไว้

ตาราง 3.18 Show Score Use Case

Extension Use Case : Show Score
Use case ID : 17
Actors : ผู้เล่น (Player)
Goal : เพิ่มตัวเลข Score เมื่อฆ่าศัตรูได้
Preconditions : เมื่อฆ่าศัตรูจะตัวเลขจะบวก 1
Postconditions : แสดง UI ของ Score เพิ่มตัวเลขทุกๆ ครั้งที่ได้ฆ่าศัตรู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.19 การออกจากเกม (Quit) Use Case

Extension Use Case : ออกจากเกม (Quit)
Use case ID : 18
Actors : ผู้เล่น (Player)
Goal : ออกจากเกม
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : ออกจากเกมและปิด Client เกม
Trigger : กดปุ่ม Quit หรือ ESC

ตาราง 3.20 Show HP Enemy Use Case

Extension Use Case : Show HP Enemy
Use case ID : 19
Actors : ศัตรู (Enemy)
Goal : มี HP Bar ของ Enemy แสดงบนหน้าจอ
Preconditions : ผู้ใช้สามารถใช้งานได้เลย
Postconditions : แสดงบน UI แบบสี่เหลี่ยมผืนผ้าด้านล่างต่อจาก HP Bar สีเหลือง
Trigger : -

ตาราง 3.21 Enemy Attack Use Case

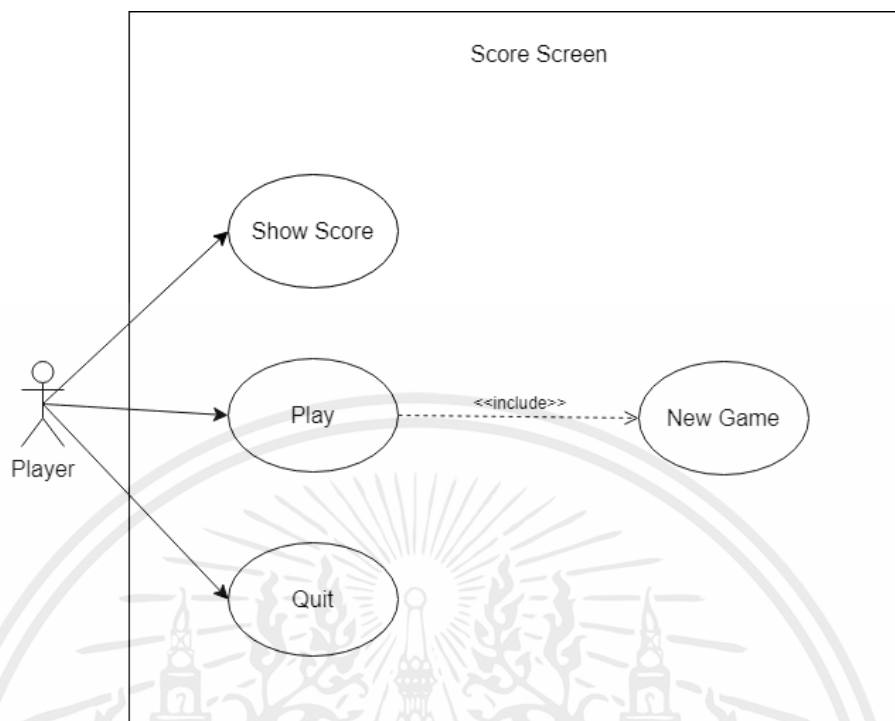
Extension Use Case : Show HP Enemy
Use case ID : 20
Actors : ศัตรู (Enemy)
Goal : แสดง Action การโจมตีของศัตรู
Preconditions : Enemy สามารถโจมตีได้
Postconditions : แสดง Action ของตัวละคร ตามที่ตั้งค่า Animator ไว้

ตาราง 3.22 Enemy Run Use Case

Extension Use Case : Run
Use case ID : 21
Actors : ศัตรู (Enemy)
Goal : แสดง Action การ โจมตีของศัตรู
Preconditions : Enemy สามารถ โจมตีได้
Postconditions : แสดง Action ของตัวละคร ตามที่ตั้งค่า Animator ไว้

ตาราง 3.23 Enemy Die Use Case

Extension Use Case : Die
Use case ID : 22
Actors : ศัตรู (Enemy)
Goal : ตายเมื่อ HP น้อยกว่าหรือเท่ากับ 0
Preconditions : ผู้ใช้สามารถแสดง Action ตายได้
Postconditions : แสดง Action ของตัวละคร ตามที่ตั้งค่า Animator ไว้



รูป 3.11 Use Case ของหน้าจอ Score Screen

โดยส่วนรายละเอียดของ Use Case ของหน้าจอ Game Screen จะแสดงดังตาราง 3.24 ถึง 3.26

Use Case Detailed Semantics Score Screen

ตาราง 3.24 แสดงผลคะแนน (Show Score) Use Case

Extension Use Case : แสดงผลคะแนน (Show Score)
Use case ID : 23
Actors : ผู้เล่น (Player)
Goal : แสดงผลคะแนนที่หน้า Score Screen
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่ตัวเลือกการจบเกม
Trigger : -

ตาราง 3.25 การเริ่มเกม (Play) Use Case

Extension Use Case : เริ่มเกม (Play)
Use case ID : 24
Actors : ผู้เล่น (Player)
Goal : เข้าสู่หน้าเริ่มเกม
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : เข้าสู่ตัวเลือกการเริ่มเกม
Trigger : กดปุ่ม Play

ตาราง 3.26 การออกจากเกม (Quit) Use Case

Extension Use Case : ออกจากเกม (Quit)
Use case ID : 25
Actors : ผู้เล่น (Player)
Goal : ออกจากเกม
Preconditions : เปิดเข้า Client ของตัวเกม
Postconditions : ออกจากเกมและปิด Client เกม
Trigger : กดปุ่ม Quit หรือ ESC

3.4 User Interface

มีการออกแบบหน้า UI ต่างๆ ไว้ จาก Use Case Diagram ที่อธิบายก่อนหน้า 4 หน้า UI ที่มีในเกม คือ Home Screen , Mode Screen , Game Screen , Score Screen ดังภาพ



รูป 3.12 UI Home Screen



รูป 3.13 UI Mode Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



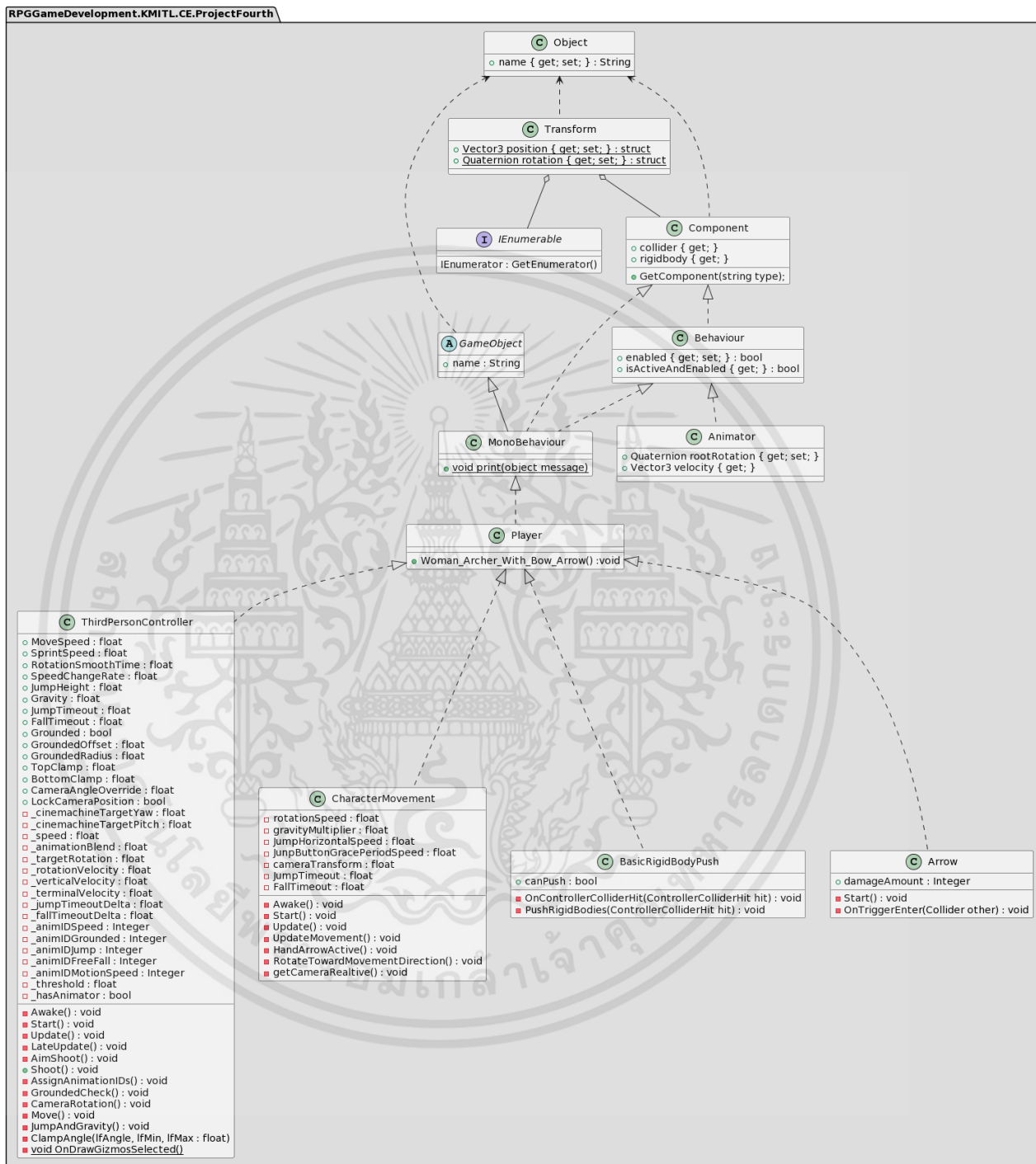
รูป 3.14 UI Game Screen



รูป 3.15 UI Score Screen

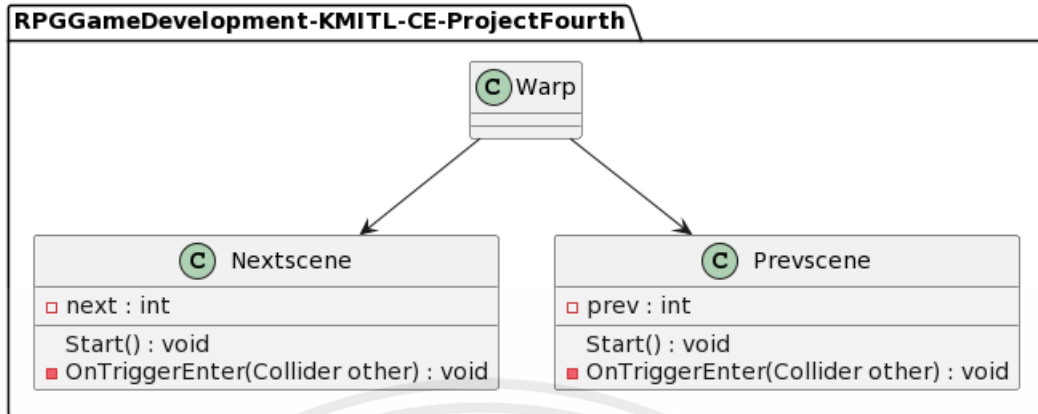
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Class Diagram

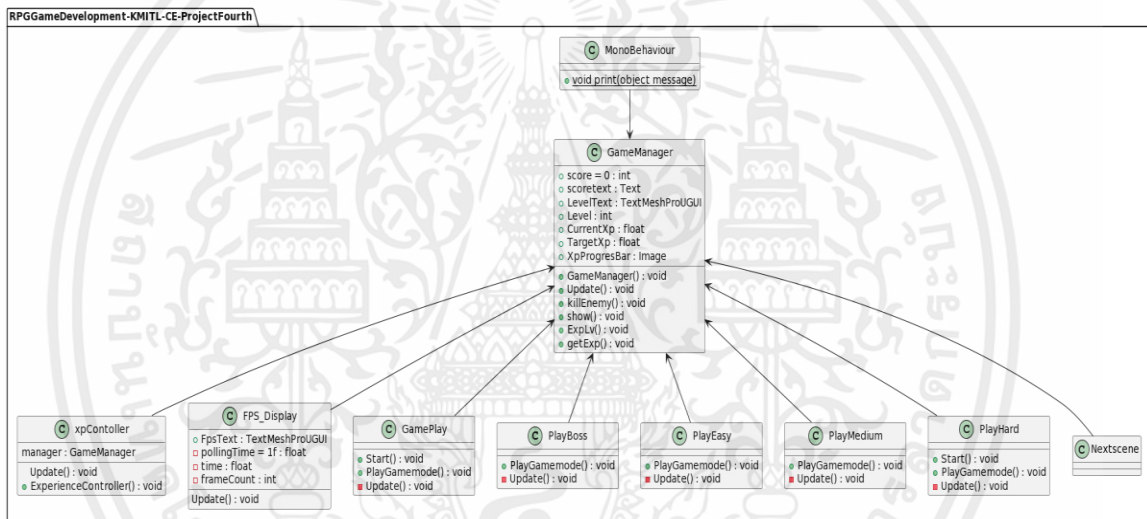


รูปที่ 3.16 Class Diagram Player

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 Class Diagram Warp



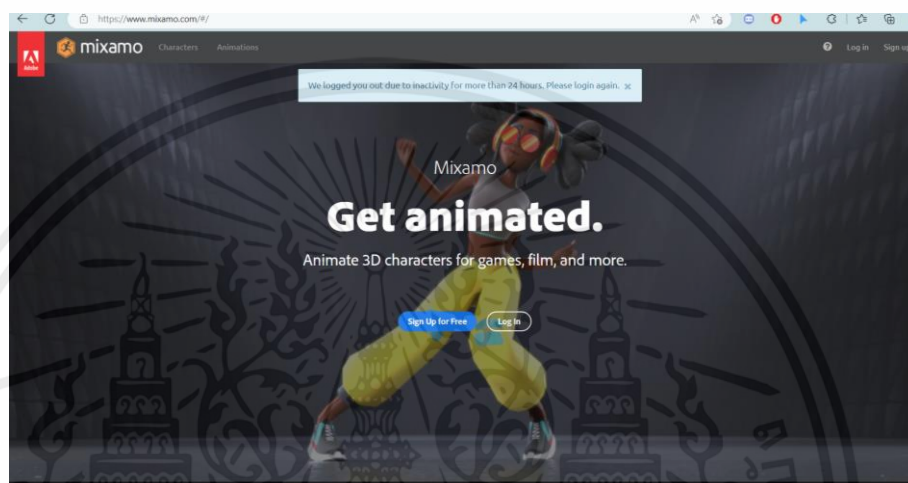
รูปที่ 3.18 Class Diagram UI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 เครื่องมือที่เกี่ยวข้อง

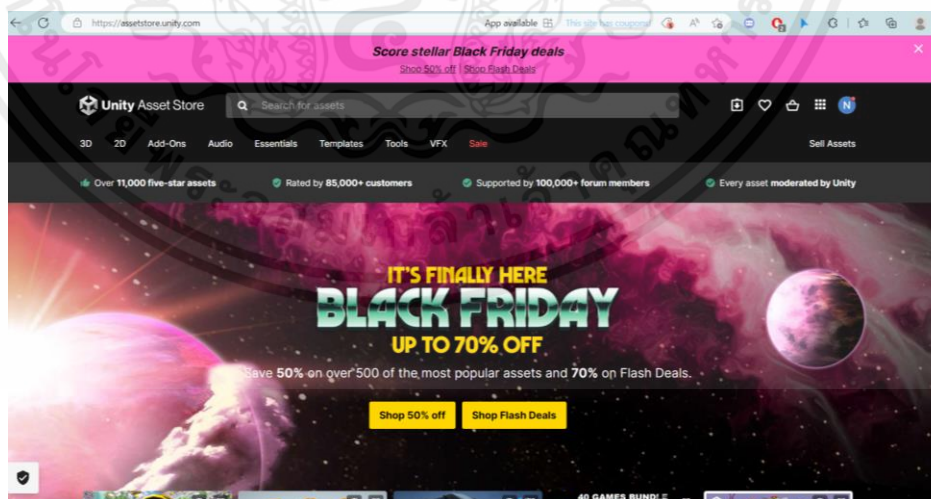
เครื่องมือที่เกี่ยวข้องในที่หมายถึง เครื่องมือหรือโปรแกรมที่นำเข้ามาเพื่อช่วยในการทำโครงการงาน

1. Maximo นำมาใช้เพื่อสะดวกต่อการพัฒนา ได้นำมาใช้ในตัวของโมเดลตัวละครทั้งชายและหญิง



รูปที่ 3.20 เว็บไซต์ Maximo

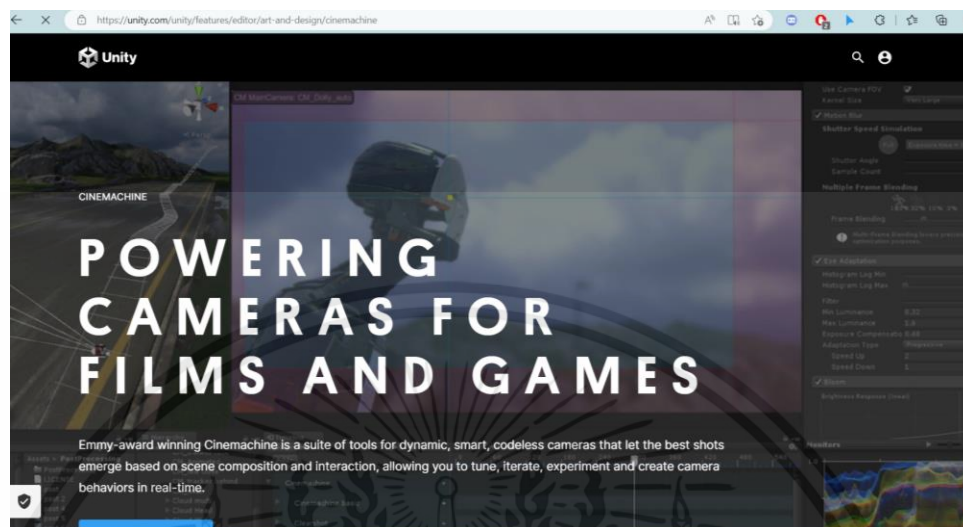
2. Asset Store ของ Unity นำมาใช้ในการออกแบบแมพและอาวุธของตัวละคร



รูปที่ 3.21 Asset Store ของ Unity

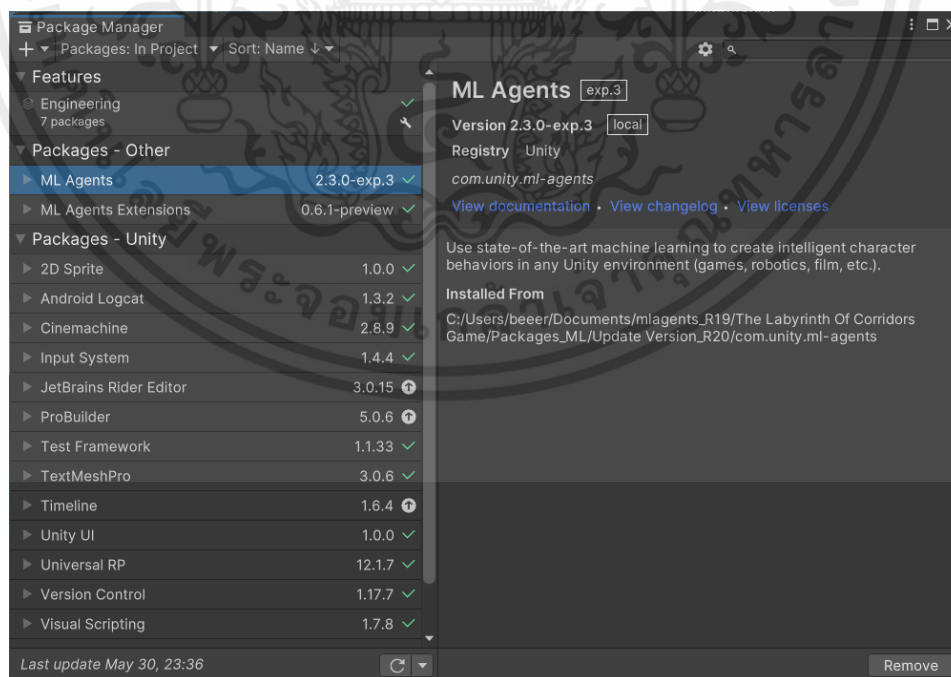
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Cine Machine Camera ใน Unity Registry นำมาใช้เพื่อมุกกล้องที่สมจริง



รูปที่ 3.22 Cine Machine Camera ของ Registry Unity

4. Machine Learning ใน ML-Agent นำมาใช้เพื่อฝึกเทรนบอสให้มีความสามารถที่เอาชนะ Player ได้

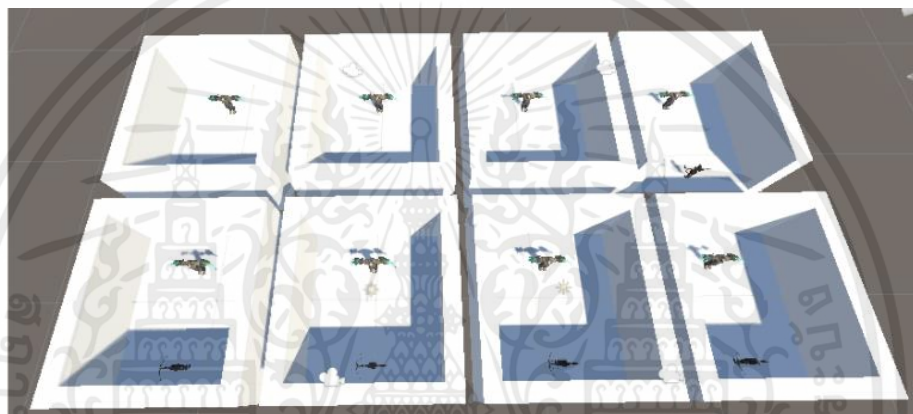


รูปที่ 3.23 Machine Learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมนี้สร้างขึ้นเพื่อควบคุมอาวุธของผู้เล่น เมื่อลูกรถถูกปล่อยออกไปให้ทำการลบทิ้งเมื่อไม่โดน Object ลบเพื่อลดความหน่วงของพื้นที่ และปรับค่าความเร็วของลูกรถได้

รูปแบบในการเทรนบอสให้ฉลาดได้ด้วยตัวเอง เราใช้ PPO หรือ Proximal Policy Optimization ในการเทรนบอส เหตุผลที่เลือกใช้ตัว ML ตัวนี้เพราะ ในสภาพแวดล้อม เราจะใช้ player 1 และ Boss 1 1:1 เพื่อให้บอสได้เรียนรู้การโจมตีจาก Player ให้เทรนไปเรื่อย ๆ จนกว่าจะชนะ เมื่อชนะ Player 1 ครั้งก็จะให้รางวัลแก่ Boss และจะเพิ่มรางวัลให้ครั้งละ 1 และสามารถใช้แมพหลาย ๆ อินสแตนซ์ multiple instances เพื่อให้การรวบรวมข้อมูลเป็นแบบคู่ขนาน parallelize และทำให้การฝึกเร็วขึ้น



รูปที่ 3.24 Training our player agent using multiple instances

บทที่ 4

ผลการดำเนินงาน

ผู้จัดทำได้ทำการทดลองเล่นตัวเกมเองเพื่อตรวจสอบการทำงานของเกมที่ตรงไปตามที่กำหนดไว้ตามขอบเขตการทำงานหรือไม่ หากมีปัญหาจะแสดงบัคออกมาผ่านการ Console ของเกมที่นอกเหนือจากระบบที่ตั้งไว้ก็จะดำเนินการแก้ไขตามขั้นตอน โดยแบ่งออกเป็น 2 ส่วน ดังนี้

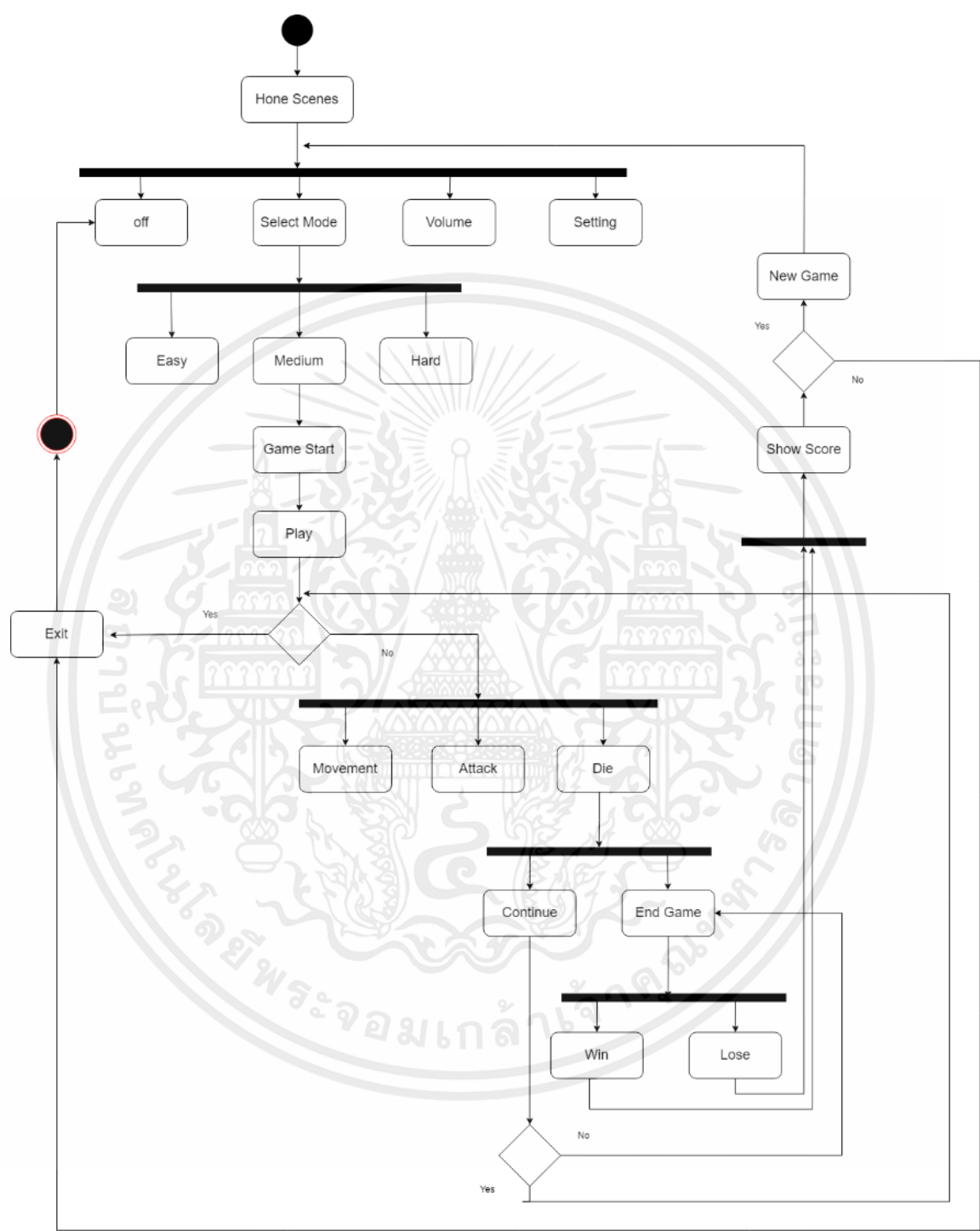
- 4.1 วิเคราะห์การพัฒนาระบบในการเล่นเกม
- 4.2 การทดลองการทำงานของตัวเกม

4.1 วิเคราะห์การพัฒนาระบบในการเล่นเกม

ในส่วนของการวิเคราะห์การพัฒนาระบบเพื่อให้ง่ายต่อขั้นตอนการดำเนินงานและทราบถึงความต้องการของระบบที่ทางผู้จัดทำได้วางขอบเขตของการทำงานไว้ว่าตรงตามเงื่อนไข ดังต่อไปนี้

1. Start เริ่มต้นการทำงาน
2. Home Scenes หน้าจอหลักเกม
3. Volume ปุ่มลดเสียง/เพิ่มเสียงของระบบ
4. Setting ปุ่มการตั้งค่า
5. Select Mode หลังจากเข้าสู่เกมจะพบหน้าจอเมนูเพื่อเลือกว่าต้องการเล่นโหมดใด
6. Choose a Mode เพื่อเลือกโหมดที่ต้องการเล่น
 - 6.1 Mode Easy สำหรับผู้เล่นเริ่มต้น
 - 6.2 Mode Medium สำหรับผู้เล่นที่เริ่มเชี่ยวชาญ
 - 6.3 Mode Hard สำหรับผู้เล่นที่เชี่ยวชาญในการเล่นเอาชนะ
7. Game Start เพื่อให้ทราบถึงการเริ่มต้นของเกม
8. Play เข้าสู่ตัวเกมพร้อมเริ่มเล่น
9. Decision ของ Play จะเข้าสู่เงื่อนไข 2 ทางเลือกว่าจะปุ่มเพื่อเล่นหรือจะกดปุ่มออกจากเกม
10. Exit ปุ่มเพื่อทำการออกเกม
11. Off เมื่อกดปุ่มออกเกมจะปิดตัวโปรแกรมหรือจบการทำงาน
12. Movement การบังคับตัวละครด้วยปุ่ม เดินหน้า W / ซ้าย A / ขวา D / หลัง S

13. Attack การบังคับตัวละครให้โจมตีได้โดยการใช้เมาส์ปุ่มขวาเพื่อลือคเป้าศัตรูและใช้เมาส์ปุ่มซ้ายเพื่อยิง โจมตีทำคามท
 14. Die เมื่อเลือด (HP) ของผู้เล่นน้อยกว่าหรือเท่ากับ 0
 15. Decision ของ Die จะเข้าสู่เงื่อนไข 2 ทางเลือกกว่า Player ชนะ หรือ Enemy ชนะ
 16. Continue ปุ่มเล่นต่อ
 17. End Game ปุ่มจบเกม
 18. Win แสดงหน้าจอชนะ
 19. Lose แสดงหน้าจอแพ้
 20. Show score โชว์คะแนนเมื่อจบเกมแล้ว
 21. Decision ของ Show score จะเข้าสู่เงื่อนไข 2 ทางเลือกกว่าจะปุ่มเพื่อเล่นเกมใหม่หรือจะออกเกม
 22. New Game ดายจะแสดงหน้าจอเพื่อเริ่มเกมใหม่
 23. End จบการทำงาน
- รูปแบบการเล่นเป็นไปตามผังรูปที่ 4.1



รูปที่ 4.1 Activity Diagram การทำงานของระบบเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองการทำงานในส่วนของตัวเกม

1. การเคลื่อนไหวกของตัวละคร
2. การโจมตีระยะใกล้ และระยะไกล
3. การใช้ Skill โจมตีของ Boss
4. AI ที่ใช้กับ Enemy
5. EXP ระบบเพิ่ม Level
6. พลังชีวิต HP
7. การตาย Die
8. Mode Easy, Medium, Hard
9. วาร์ป Warp
10. การทำ Damage
11. Action การเคลื่อนไหวกตัวละครและ Action อื่นๆ
12. UI ต่างๆ ภายในเกม
13. Reinforcement Learning (RL).
14. Minimap
15. Score

4.2.1 การเคลื่อนไหวกของตัวละคร

1. จุดประสงค์ของการทดลอง

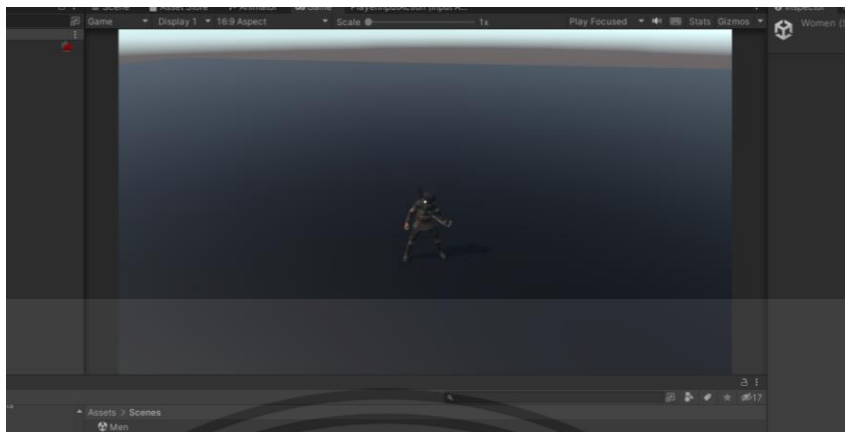
เพื่อทดสอบการเดินทางหน้า-หลัง-ซ้าย-ขวา กระโดด และมูมกลิ้ง ว่าสามารถควบคุมผ่านคีย์บอร์ดและเมาส์ได้

2. วิธีการทดลอง

ทดลองโดยการสร้าง Terrain ขึ้นมาเป็นพื้นที่ให้ยืนเพื่อให้ผู้เล่นมีพื้นที่ในการเคลื่อนไหวก แล้วทดลองโดยการกดปุ่ม W,A,S,D ว่าสามารถควบคุมตัวละครได้ไหม และทำการหมุนเมาส์เพื่อทดสอบว่ามูมกลิ้งมูมตามเมาส์ได้หรือไม่

3. ผลการทดลอง

สามารถควบคุมตัวละครผ่านคีย์บอร์ดและสามารถควบคุมมูมกลิ้งผ่านการเคลื่อนไหวกของเมาส์ได้ดังรูป 4.2



รูปที่ 4.2 ทดสอบการเคลื่อนที่ของตัวละครและการหมุนมุกกล้อง

4.2.2 การโจมตีของระยะใกล้ และระยะไกล

1. จุดประสงค์ของการทดลอง

เพื่อทดสอบการระยาระยะการโจมตีของตัวผู้เล่น ว่าสามารถทำการโจมตีได้

2. วิธีการทดลอง

ทดลองการโจมตีระยะไกลโดยการสร้าง Object ขึ้นมาเป็นก่อกองที่เหลี่ยม เพื่อใช้แทนตัว Enemy แล้วทำการคลิกขวาเพื่อเล็ง

3. ผลการทดลอง

สามารถคลิกขวาเพื่อเล็งและคลิกซ้ายเพื่อยิงลูกธนูออกไปได้ในการโจมตีของตัวละครหญิงระยะไกลดังรูป 4.3



รูปที่ 4.3 ทดสอบการโจมตีระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ศัตรูและการโจมตี

1. จุดประสงค์ของการทดลอง

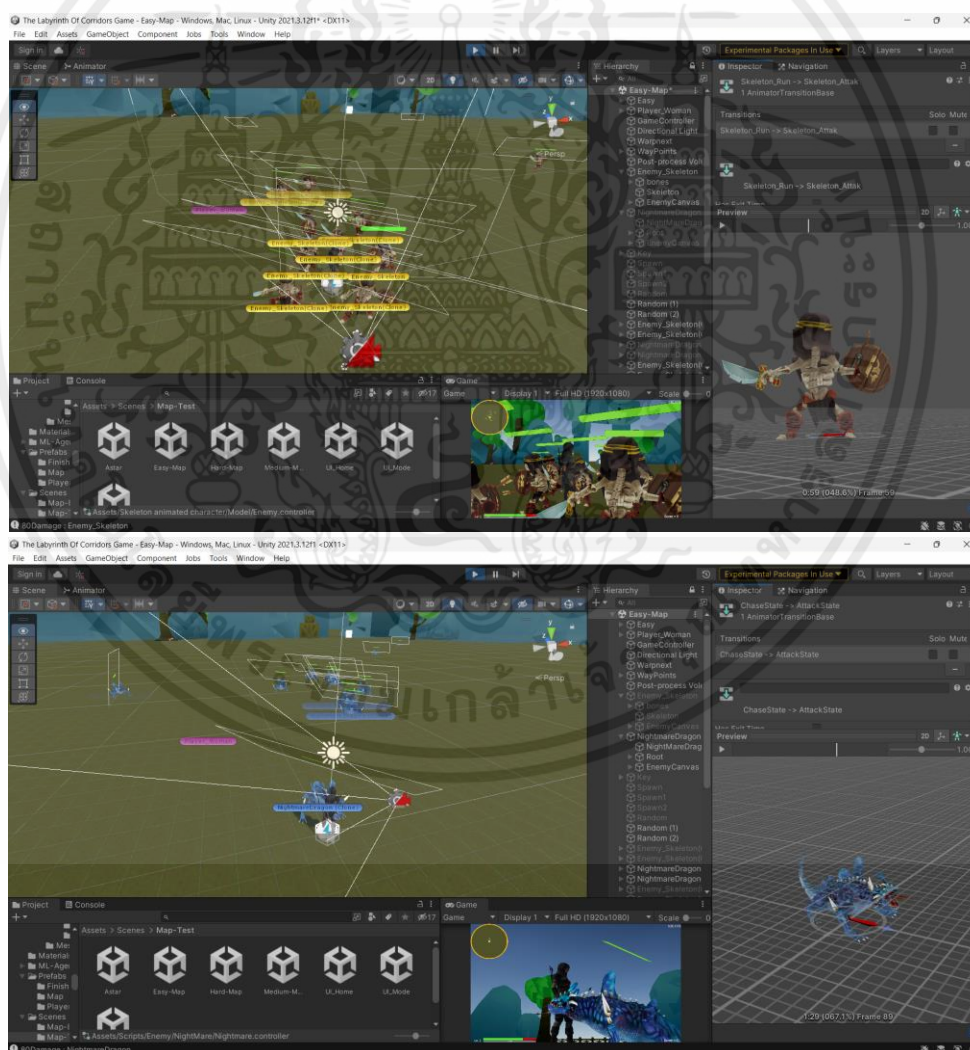
เพื่อทำการทดสอบการเดินหน้า-หลัง-ซ้าย-ขวา และการโจมตีของตัวละคร

2. วิธีการทดลอง

ทดลองโดยการสร้าง Terrain ขึ้นมาและนำโมเดลตัวละครมาทดสอบบนพื้นที่ว่าตอน Test Run ตัวเกมไม่เกิดปัญหาพร้อมทั้งกำหนดพารามิเตอร์บังคับตัวละครให้โจมตีเมื่อเจอ Player

3. ผลการทดลอง

ตัวละครศัตรูทั้ง 2 แบบ สามารถเคลื่อนไหวและโจมตีได้ ดังรูป



รูปที่ 4.4 การโจมตีของ Enemy_Skeleton และ NightmareDragon

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การใช้ Skill โจมตีของ Boss

1. จุดประสงค์ของการทดลอง

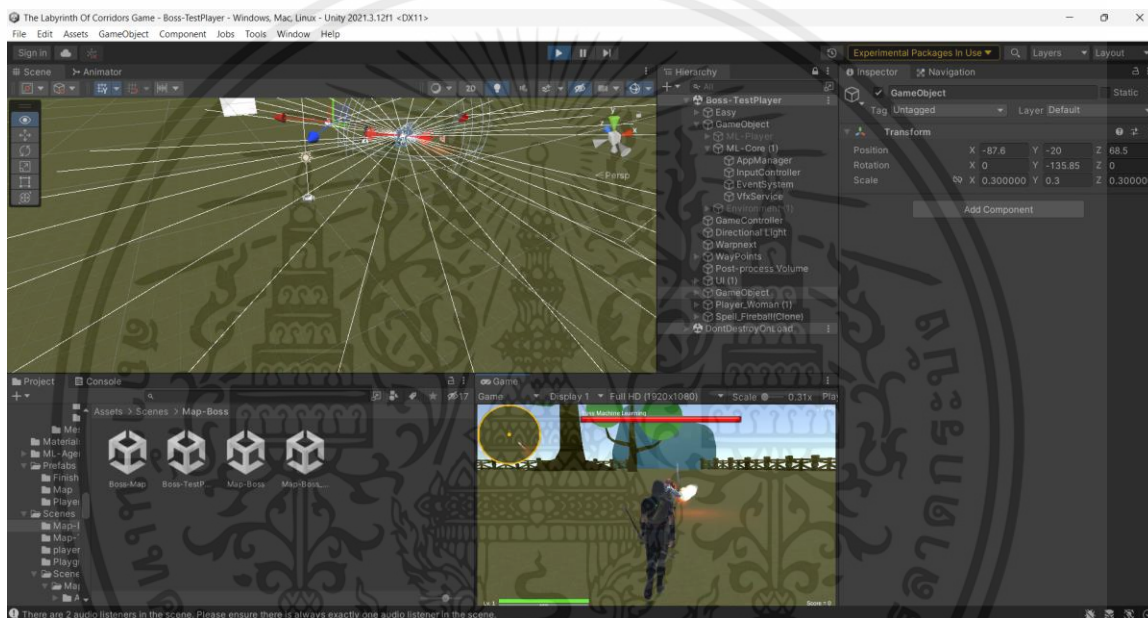
เพื่อทดสอบการ โจมตีของ Boss แบบใช้สกิลลูกไฟออกมา

2. วิธีการทดลอง

ทดสอบโดยการเริ่มเล่นแล้วให้ Boss ลงโจมตีโดยใช้แต่สกิล

1. ผลการทดลอง

Boss สามารถโจมตีโดยใช้สกิล ได้ดังรูป 4.5



รูปที่ 4.5 Skill การโจมตีของ Boss

4.2.5 AI ที่ใช้กับ Enemy

1. จุดประสงค์ของการทดลอง

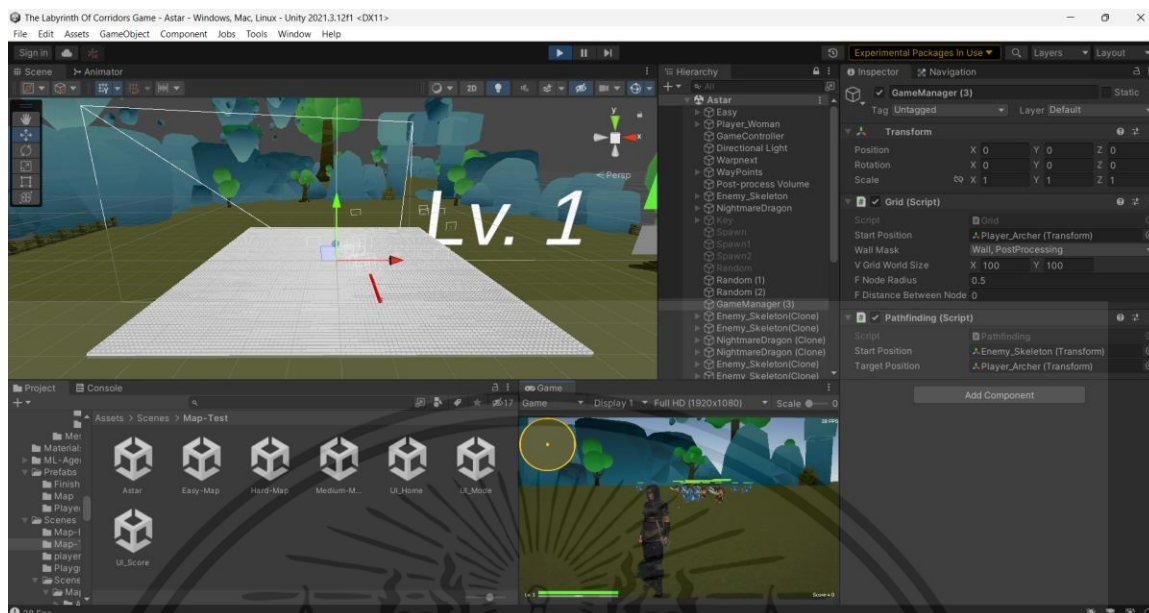
เพื่อทดสอบ AI ของ Enemy ในการค้นหาเส้นทางที่สั้นที่สุดเพื่อไปหา Player

2. วิธีการทดลอง

สร้าง Script การเช็คเส้นทางของ Enemy ระหว่างเดินไปหา Player แล้วกำหนดให้เส้นทางข้างหน้าที่จะเดินเป็นเส้นสีแดงเพื่อทดสอบว่า Enemy สามารถไปหา Player ได้จริง

3. ผลการทดลอง

Enemy สามารถ เดินไปยังเส้นทางที่สั้นที่สุดเพื่อไปหา Player ได้จริงดังรูป 4.6



รูปที่ 4.6 Enemy AI A* PathFinding

4.2.6 EXP ระบบเพิ่ม Level

1. จุดประสงค์ของการทดลอง

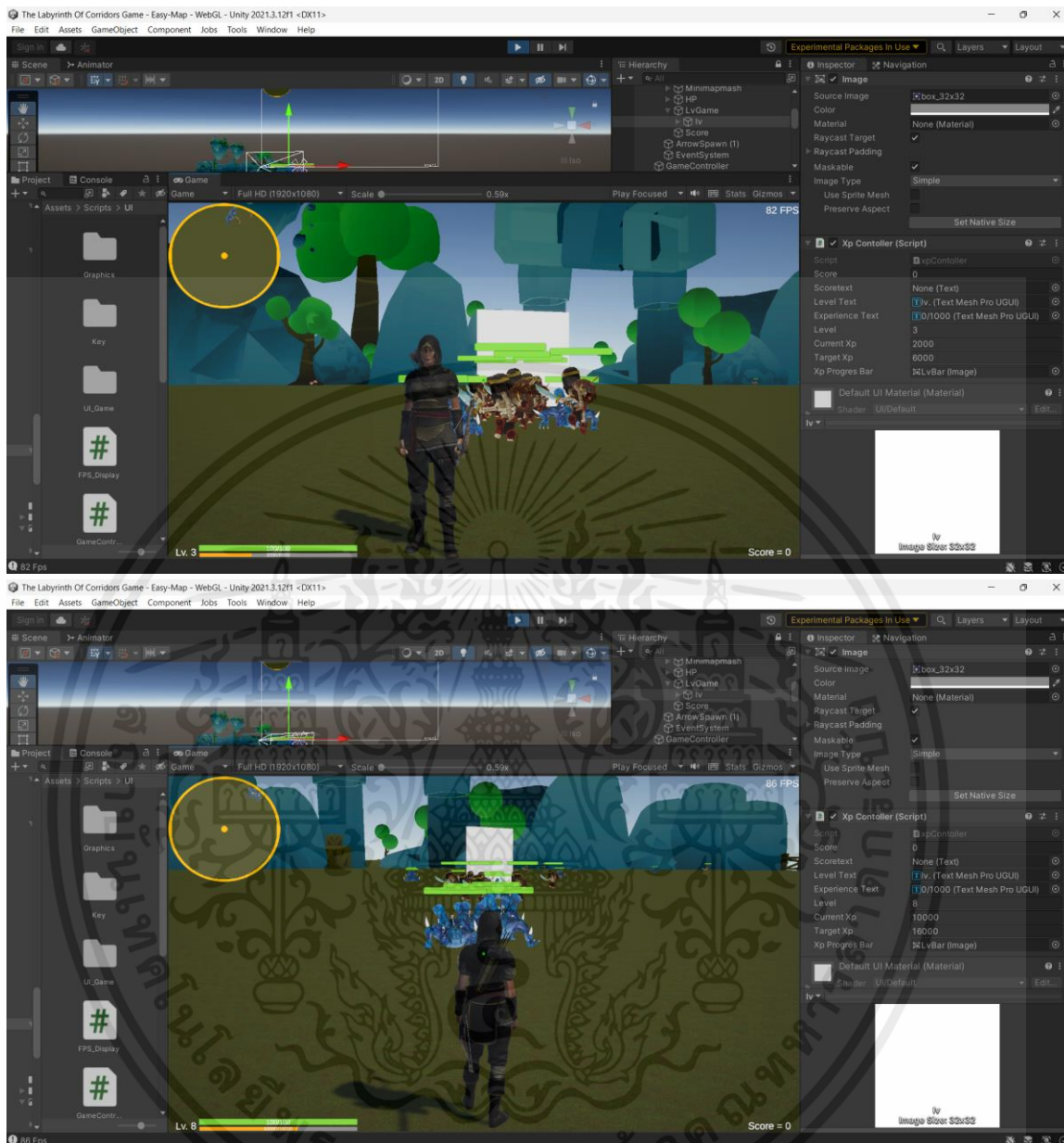
เพื่อให้เวลาที่เอาชนะศัตรูภายในพื้นที่หรือฆ่าศัตรูได้จะได้รับ Exp เพิ่มและจะ Level เพิ่ม ตามเงื่อนไขการเลื่อนระดับของ Level

2. วิธีการทดลอง

ทดสอบโดยการฆ่าศัตรูและสังเกตที่ UI Bar EXP หรือหลอดสีเหลืองว่าเพิ่มขึ้นและเปลี่ยนตัวเลข จาก 0/2000 เป็น 0/4000 เมื่อเลื่อนระดับ Level เป็น Level 2 จาก Level 1 เริ่มต้น

3. ผลการทดลอง

สามารถเลื่อนระดับ Level ของตัวละครได้และสามารถเพิ่มความยากในการไป Level ถัดไปทุกๆ ครั้งที่มีการเพิ่ม Level จะเพิ่ม Max Exp ที่ต้องเก็บจากการฆ่าศัตรู เพิ่มขึ้น Lv. ละ 2000 ดังรูป 4.7



รูปที่ 4.7 ระบบ EXP และระบบ Level ของผู้เล่น

4.2.7 พลังชีวิต HP

1. จุดประสงค์ของการทดลอง

เพื่อทดสอบว่าระบบ HP สามารถทำงานบนจอแสดงผลได้

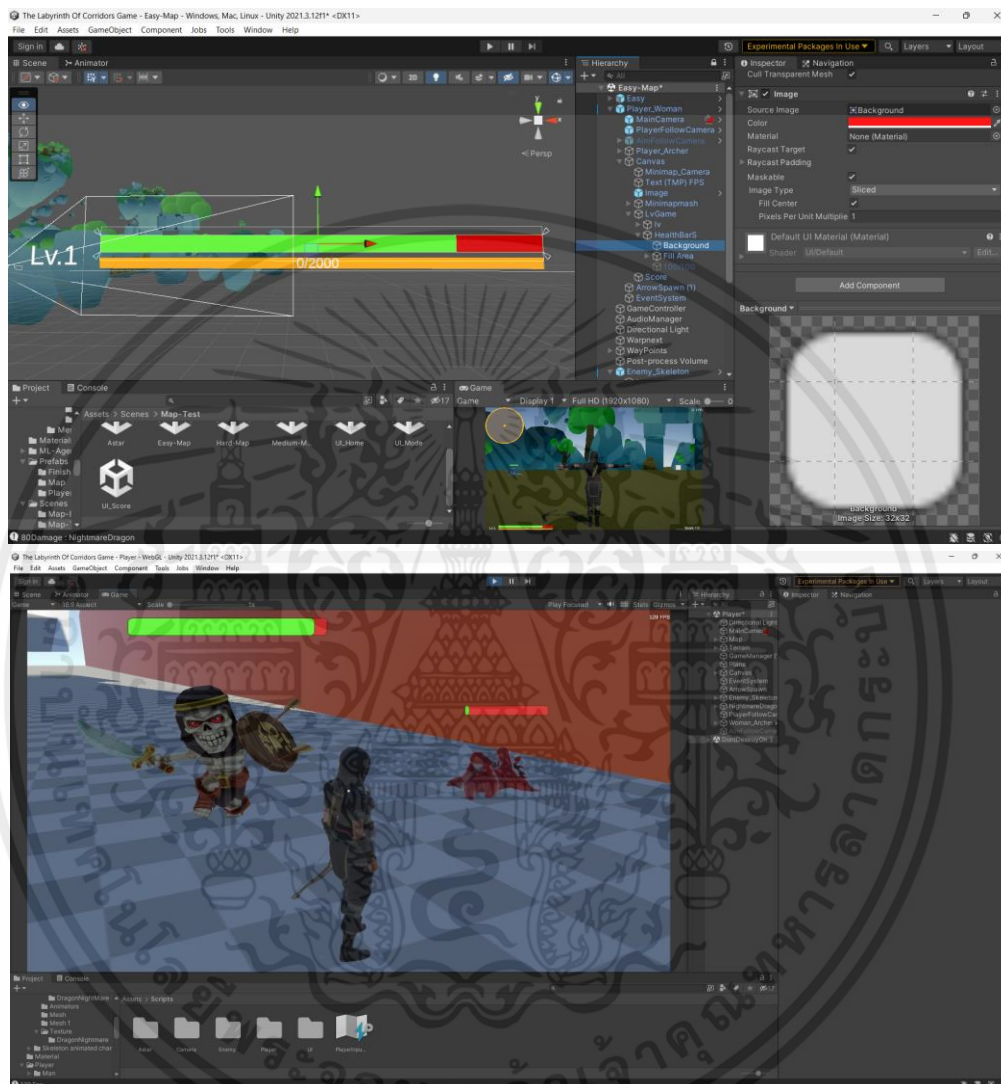
2. วิธีการทดลอง

ทำการโจมตีใส่ Enemy แล้วดูที่ HP Bar ว่าเกิดการเปลี่ยนแปลงเมื่อโดน โจมตีหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ผลการทดลอง

เมื่อโจมตีหรือสร้างคาเมทาให้กับ Enemy ส่วนของ HP Bar จะเปลี่ยนจากสีเขียวเป็นสีแดงลดตามคาเมทาที่สามารถทำได้ดังรูป 4.8



รูปที่ 4.8 ทดสอบการทำ Damage ให้กับตัว Enemy

4.2.8 การตาย Die

1. จุดประสงค์ของการทดลอง

เพื่อทดสอบการตายของตัวละครทั้ง Player และ Enemy

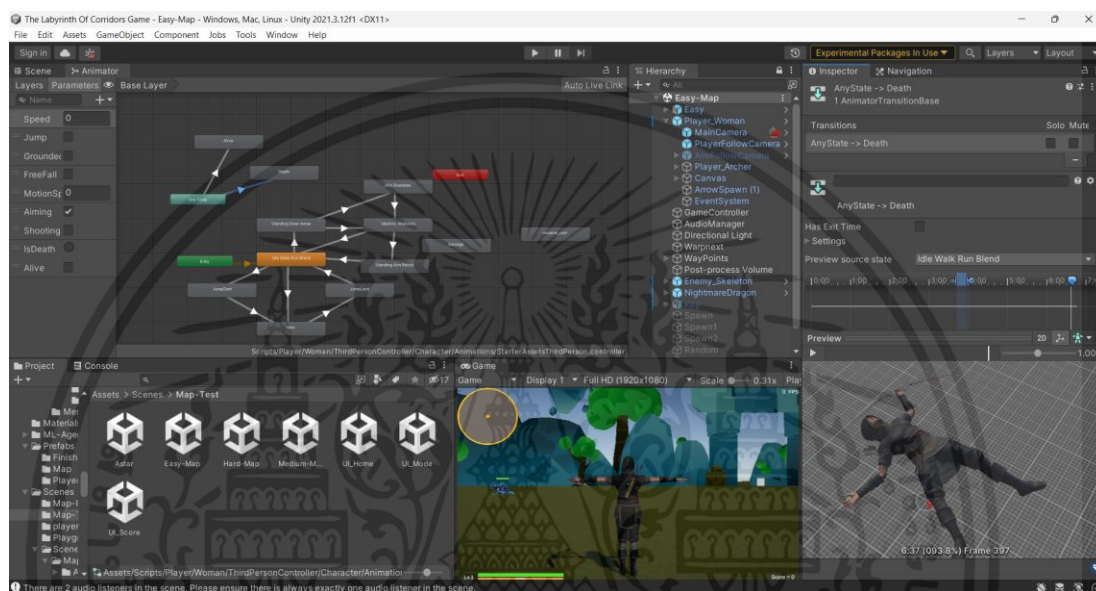
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วิธีการทดลอง

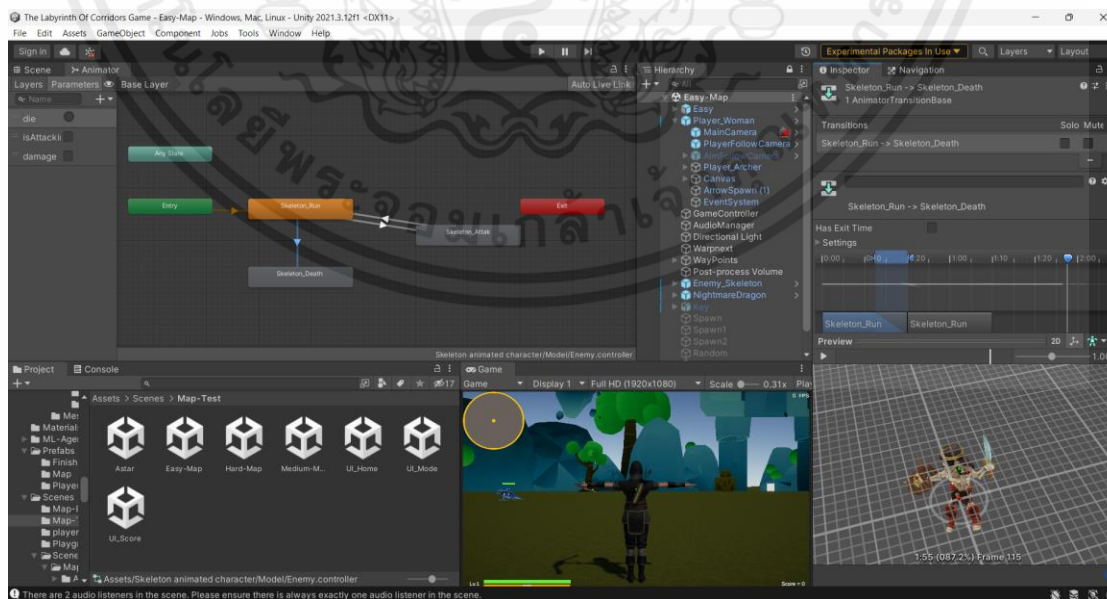
ทดสอบโดยการเพิ่ม Animation การตายให้กับตัวละครและนำไปใช้ในระบบ

3. ผลการทดลอง

จากการทดลองตัวของ Player และ Enemy สามารถตายได้ดังรูป 4.9 และ 4.10



รูปที่ 4.9 การตาย Die ของ Player



รูปที่ 4.10 การตาย Die ของ Enemy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.9 Mode Easy, Medium, Hard

1. จุดประสงค์ของการทดลอง

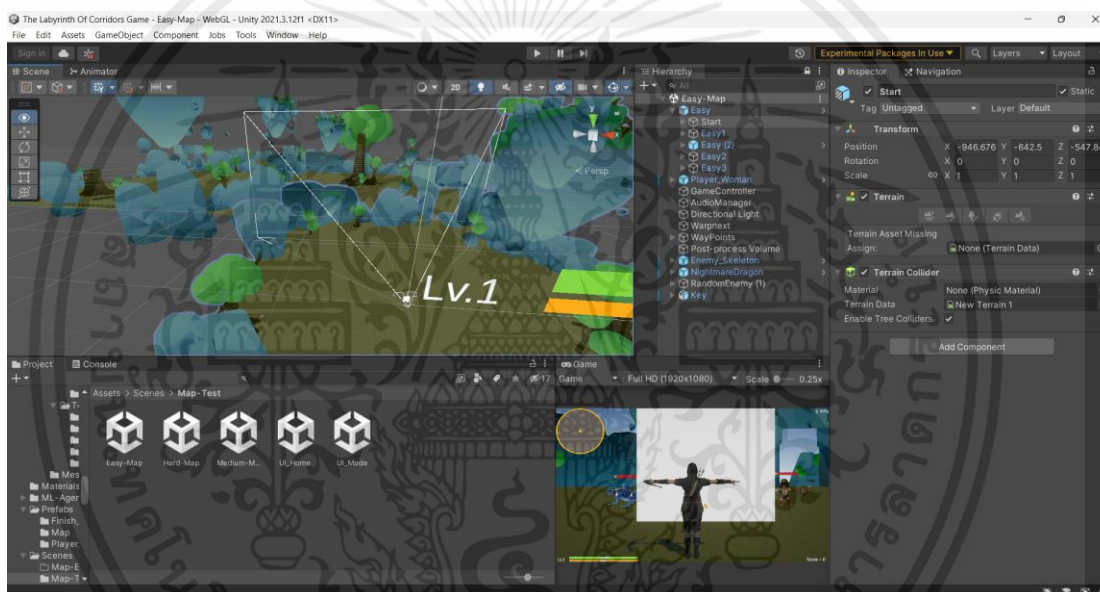
เพื่อให้สามารถเล่นในโหมดที่หลากหลายได้ผู้จัดทำจึงแบ่งเป็น 3 โหมดในการเล่น

2. วิธีการทดลอง

ทดสอบโดยการนำ Player เข้ามาใช้งานในพื้นที่ที่ได้ทำการออกแบบมาทั้ง 3 แบบ

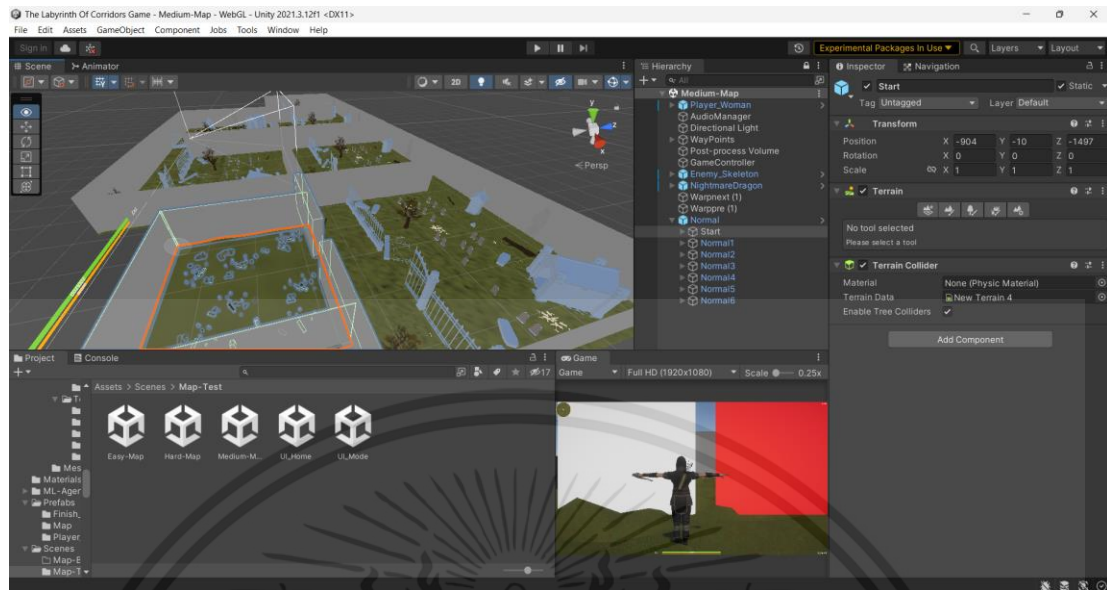
3. ผลการทดลอง

สามารถเล่นและเดินภายในแผนที่สร้างได้และไม่เกิดปัญหา

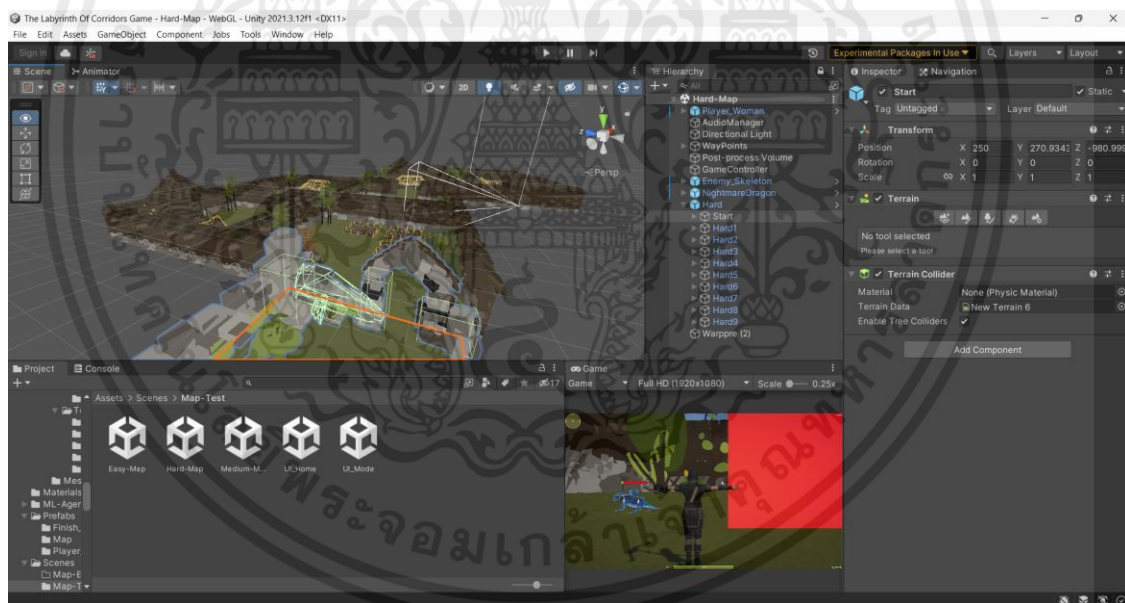


รูปที่ 4.11 Scenes Mode Easy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 Scenes Mode Medium, Hard



รูปที่ 4.13 Scenes Mode Hard

4.2.10 วาร์ป Warp

1. จุดประสงค์ของการทดลอง

เพื่อทดสอบระบบวาร์ปของการเข้าด่านว่าสามารถทำงานได้ปกติ

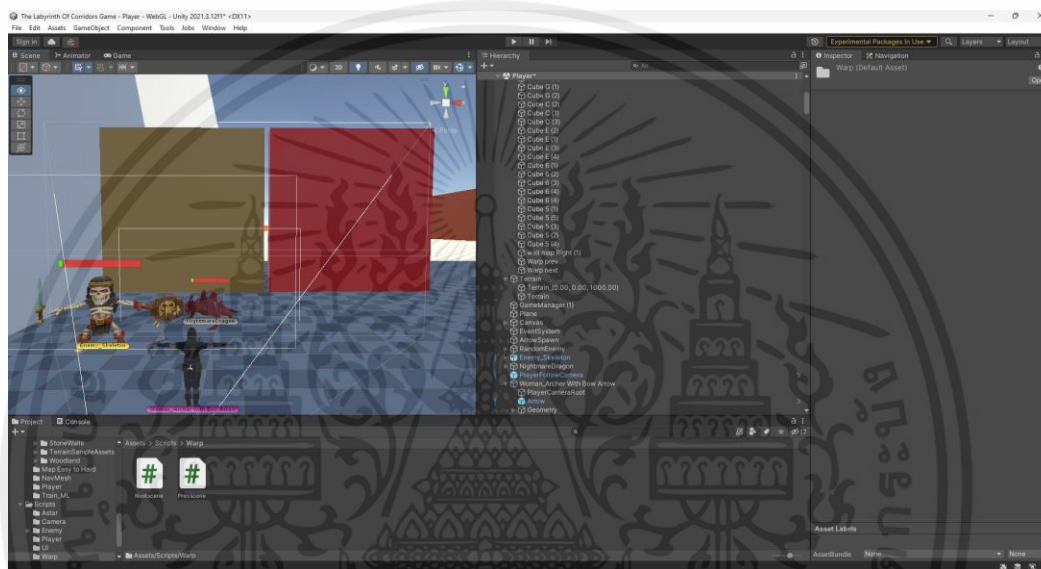
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วิธีการทดลอง

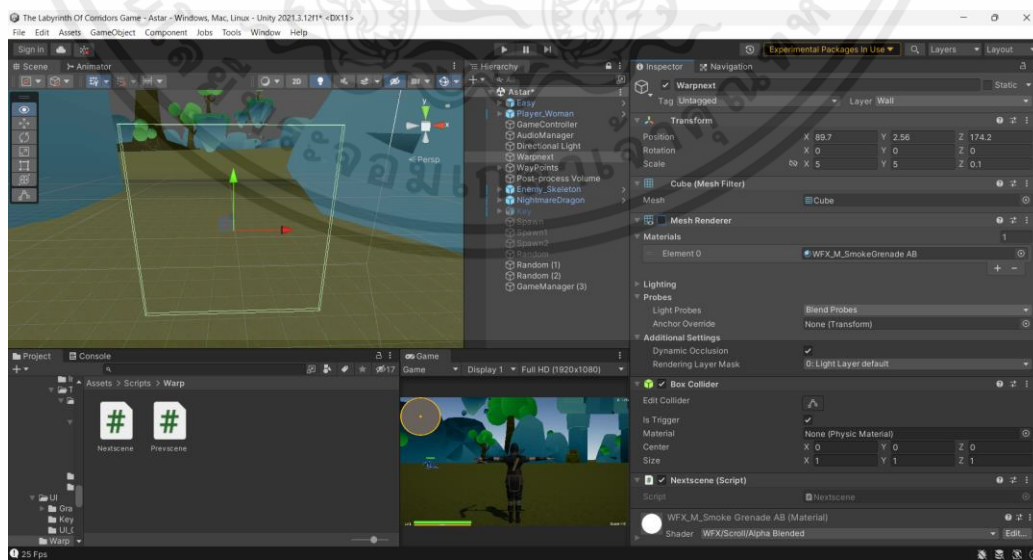
ทำการสร้างประตูลาร์ปขึ้นมา 2 ประตู ประตูสีเหลืองคือไปด้านต่อไป ประตูสีแดงคือย้อนกลับมา แล้วทำการเดินเข้าประตูไป ถ้าไปโผล่อีกด้านก็สามารถทำงานได้

3. ผลการทดลอง

สามารถทำการวาร์ปไปยัง Scenes อื่นได้ปกติ



รูปที่ 4.14 ทดสอบการวาร์ป



รูปที่ 4.15 ทดสอบการวาร์ปโดยทำให้จุดวาร์ปส่องหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.11 การทำ Damage

1. จุดประสงค์ของการทดลอง

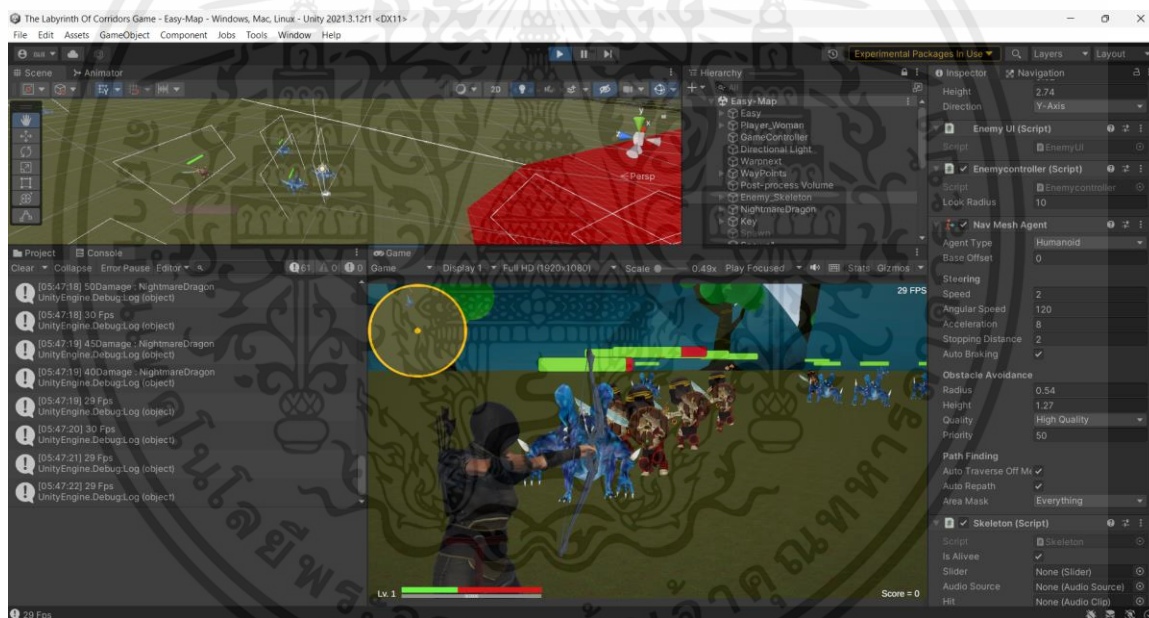
เพื่อทดสอบว่าการโจมตีของผู้เล่นสามารถสร้าง Damage ให้กับ Enemy ได้

2. วิธีการทดลอง

ทดสอบโดยการกดยิงโจมตีใส่ตัวละครทั้งด้านของ Player และ Enemy

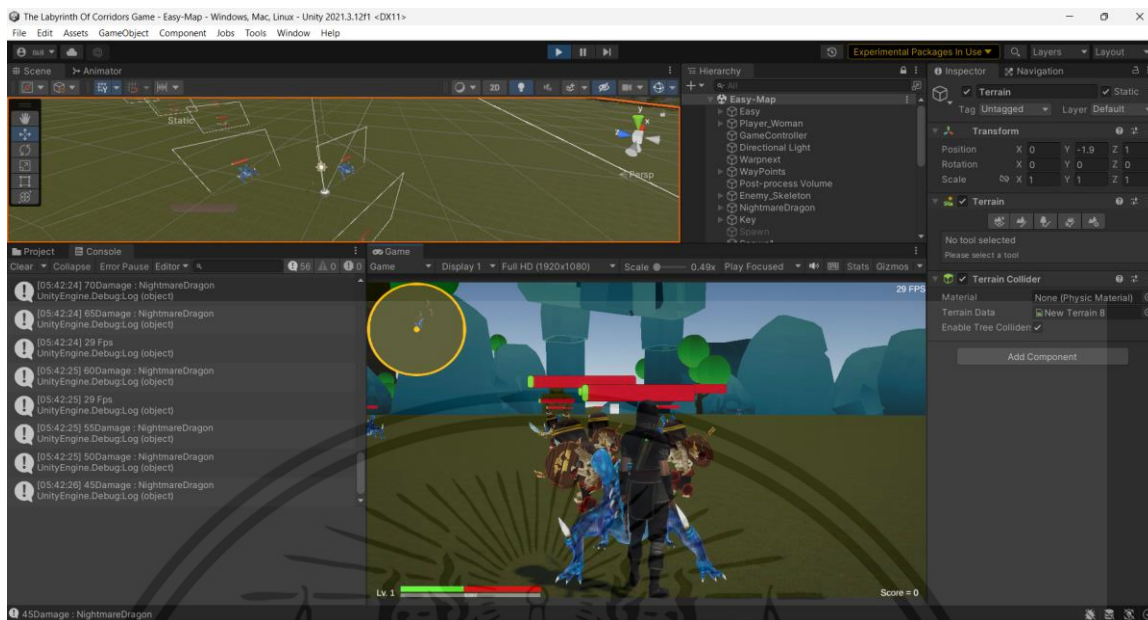
3. ผลการทดลอง

สามารถทำตามที่ได้ตัวละครได้ทั้งด้านของ Player และ Enemy และ Debug ดูข้อผิดพลาดเพื่อค้นหาที่อาจจะเกิดขึ้น โดยการแสดงค่าเมทริกซ์ที่ได้รับเวลาที่ Player ทำการโจมตีใส่ Enemy ดังรูป 4.15 และ 4.16



รูปที่ 4.16 การทำ Damage ของ Player

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 การทำ Damage ของ Enemy

4.2.12 Action การเคลื่อนไหวตัวละครและ Action อื่นๆ

1. จุดประสงค์ของการทดลอง

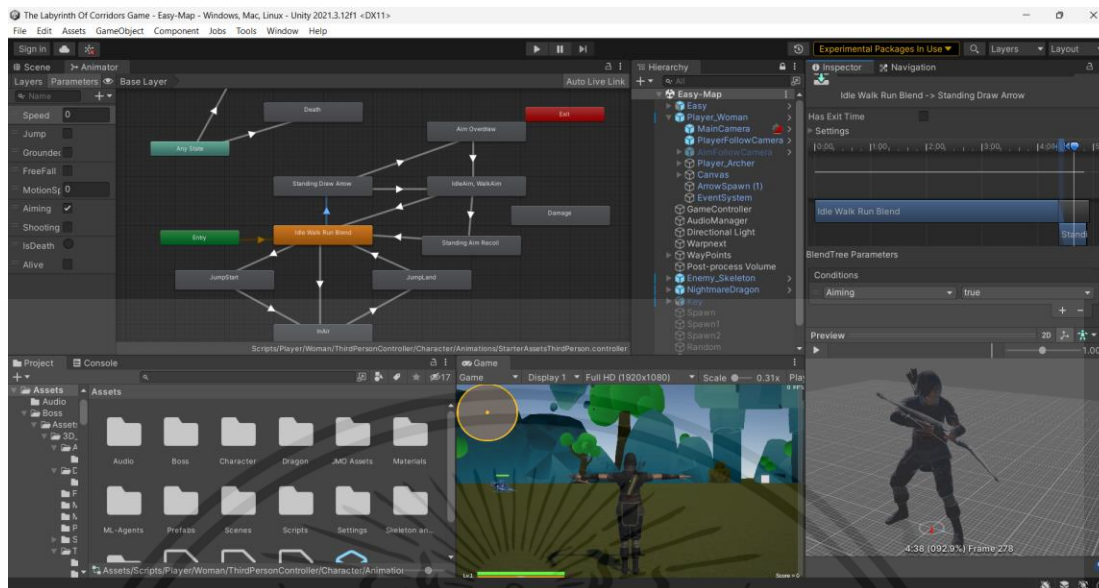
เพื่อดูการทำงานของโมเดลให้แสดงการเคลื่อนไหวปกติและตามที่ต้องการ

2. วิธีการทดลอง

ทดสอบโดยการดู time frame ของตัวละครที่ต้องการและแก้ไขท่าทางที่ต้องการให้แสดงผลในระบะเวลานั้นรวมถึงความเร็วในการแสดง Action ของตัวละครต่างๆ

3. ผลการทดลอง

สามารถกำหนดการแสดงผล Action ของตัวละครตามที่ต้องการได้โดยไม่เกิดปัญหาจากการแก้ไขส่วนของ Animator และ Inspector ดังรูป 4.17

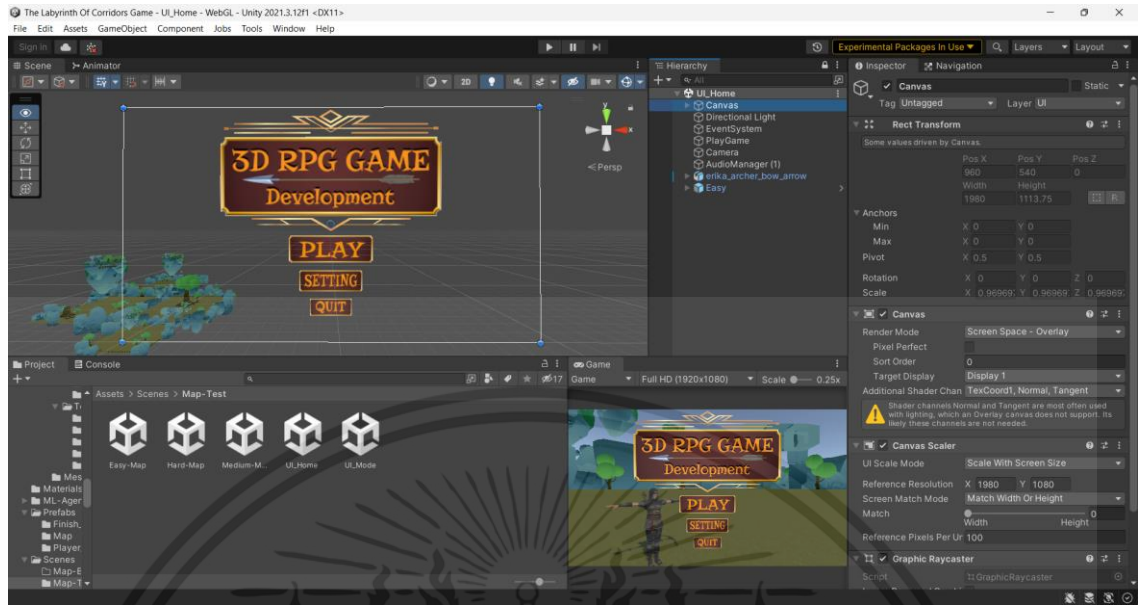


รูปที่ 4.18 Action การเคลื่อนไหวตัวละครและ Action อื่นๆ

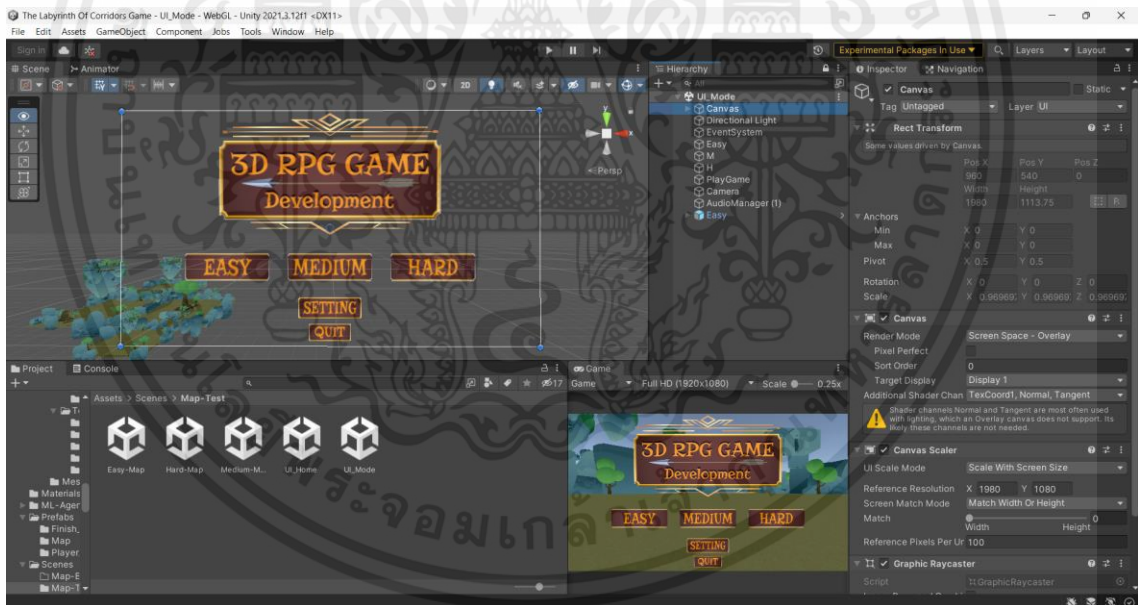
4.2.13 UI ต่างๆ ภายในเกม

1. จุดประสงค์ของการทดลอง
เพื่อให้สามารถกดปุ่มเพื่อเล่นโหมดต่างๆ ได้ รวมถึงปุ่มอื่นๆ ภายในเกม และ Interface
2. วิธีการทดลอง
ทดสอบโดยการคลิกปุ่มที่ออกแบบขึ้นมาและดูผลลัพธ์ตามที่ต้องการ
3. ผลการทดลอง
สามารถกดได้ทุกปุ่มตามปกติตามที่ออกแบบไว้ ดังรูป 4.18 ถึง 4.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

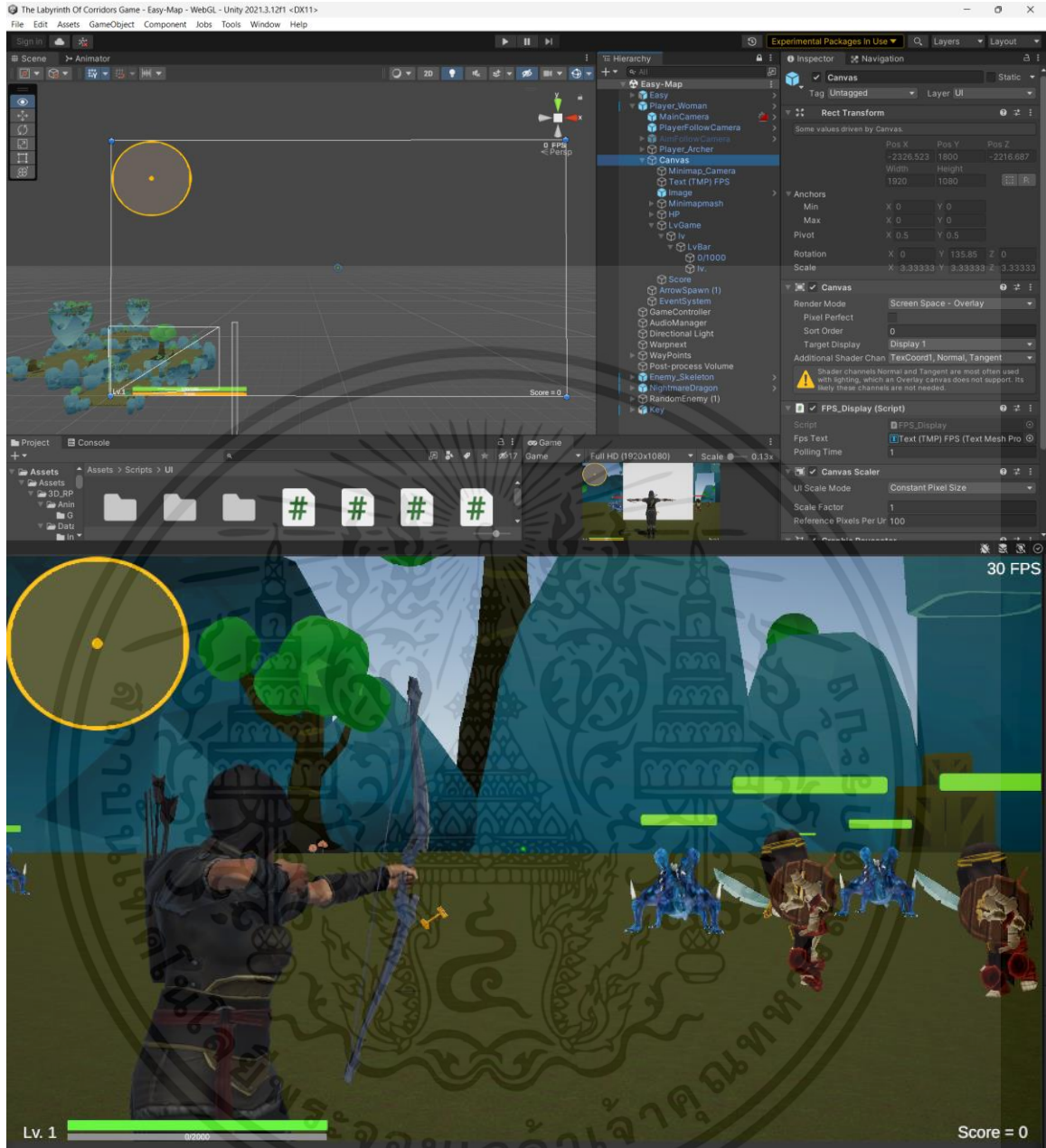


รูปที่ 4.19 UI Home



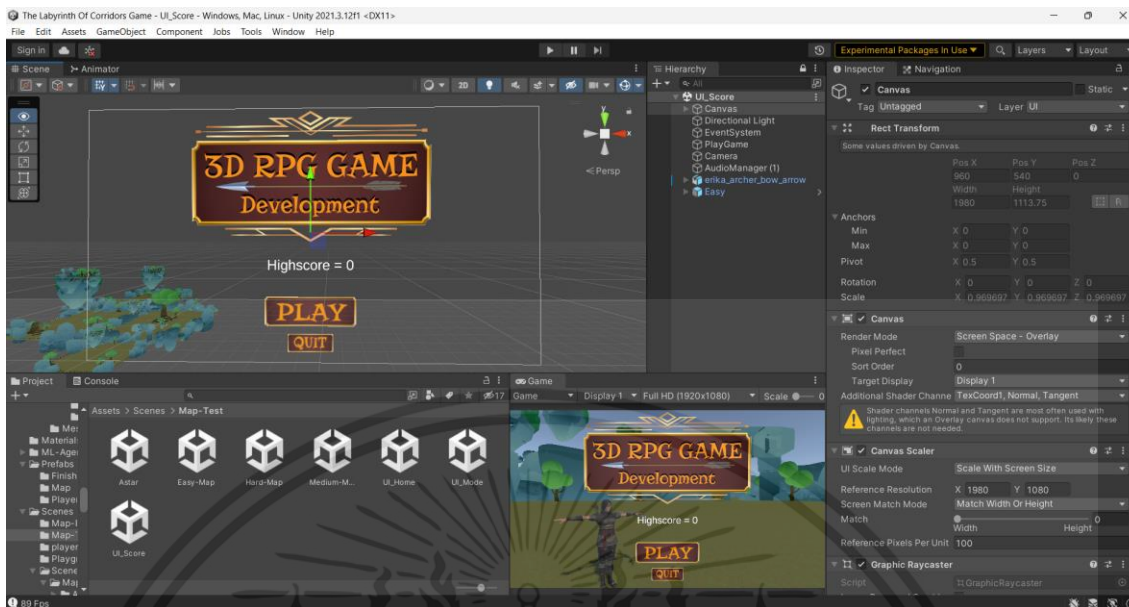
รูปที่ 4.20 UI Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 UI Game

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 UI Score

4.2.14 Reinforcement Learning (RL).

1. จุดประสงค์ของการทดลอง

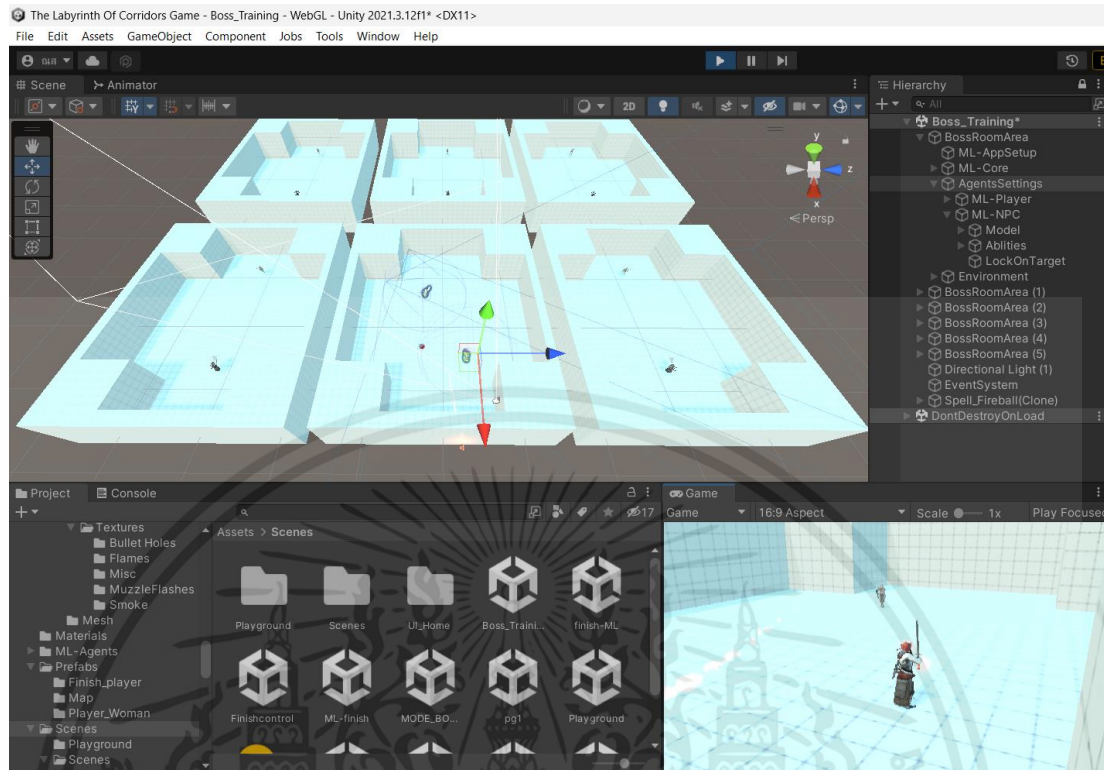
เพื่อให้ Boss สามารถ หลบหลีกและเอาชนะ Player ได้จากการเพิ่มระบบ RL

2. วิธีการทดลอง

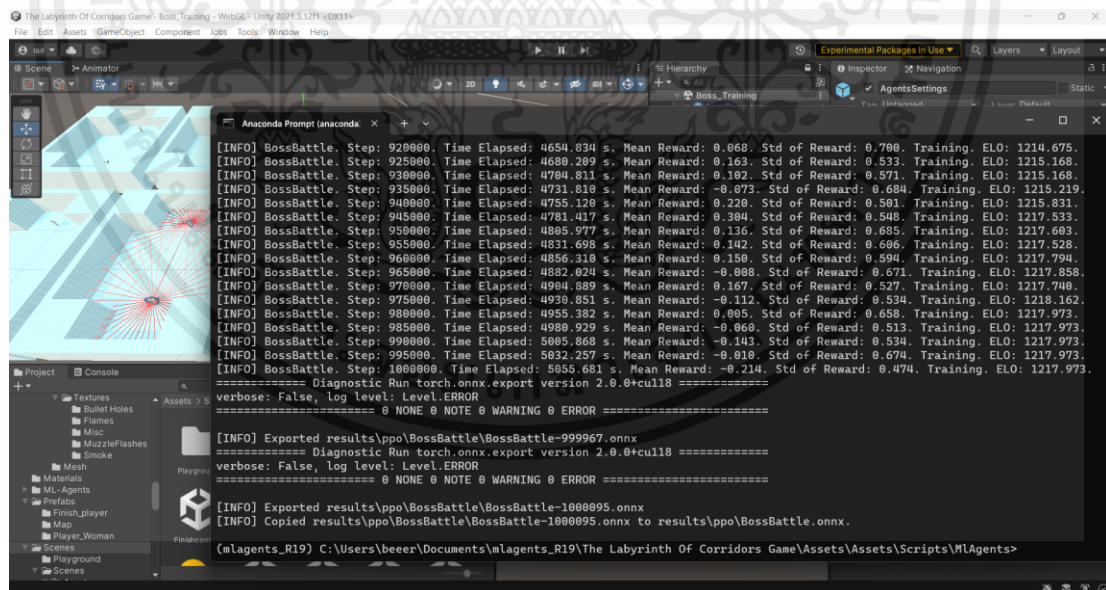
ทดสอบโดยการเริ่มเกมจากการเพิ่มไฟล์ Result เข้าตัวเกมและลองเล่น

3. ผลการทดลอง

ความสามารถของ Boss ยังไม่สามารถเอาชนะ Player ได้จริงเนื่องจากไฟล์การ Training ของ Boss ระยะเวลาฝึกฝนการทดสอบระบบยังไม่เพียงพอต่อการใช้งาน ได้จริง Boss ยังไม่สามารถเอาชนะ Player ได้ และยังหลบหลีกไม่ไปตามที่คาดหวังจากผลลัพธ์ที่นำมาใช้ ดังรูป 4.22 ถึง 4.24

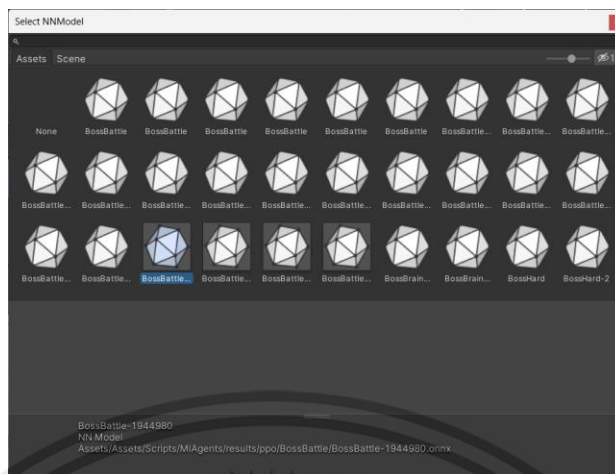


รูปที่ 4.23 การแบ่ง Training Boss หลายๆ Object เพื่อทดสอบผลลัพธ์



รูปที่ 4.24 ผลลัพธ์การ Training ผ่านการ config Anaconda

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ผลลัพธ์ไฟล์ Result RL

4.2.15 Minimap

1. จุดประสงค์ของการทดลอง

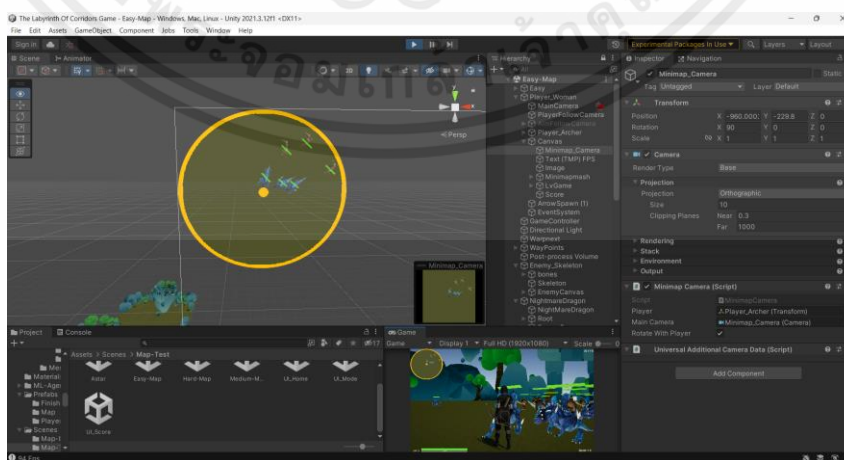
เพื่อให้ผู้เล่นสามารถดูศัตรูที่อยู่ใกล้หรือมุมมองในรูปแบบแผนที่ได้

2. วิธีการทดลอง

ทดสอบโดยการลองเล่นจริงและดูตัวละคร รวมถึง Object ทั้งหมดในเกมว่าสามารถเห็นหรือมองผ่าน Minimap ได้

3. ผลการทดลอง

สามารถดูตัวละคร รวมถึง Object ทั้งหมดในเกมได้ ดังรูป 4.25



รูปที่ 4.26 Minimap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.16 Score

1. จุดประสงค์ของการทดลอง

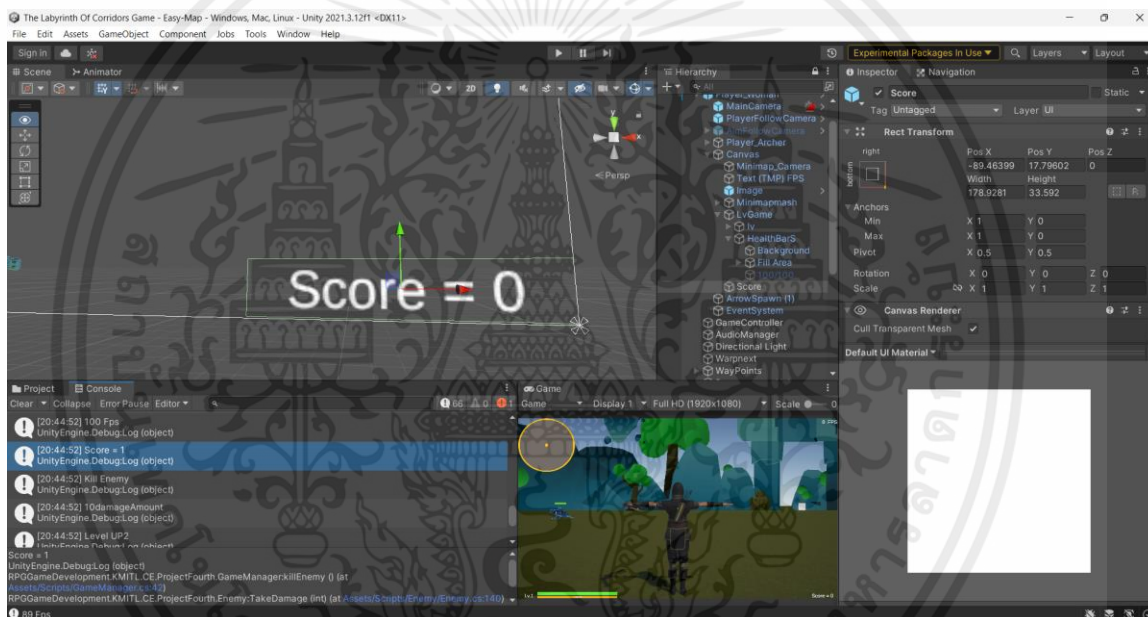
เพื่อทดสอบการแสดงผลคะแนน Score เมื่อทำการฆ่าศัตรูลงได้

2. วิธีการทดลอง

ทดลองการโจมตีใส่ศัตรูและเมื่อศัตรูตายจะเพิ่มคะแนน Score ครั้งละ 1 แต้ม

3. ผลการทดลอง

สามารถแสดงผลเพิ่มคะแนน Score ของ Player ได้ทุกๆ ครั้งที่ฆ่าศัตรู 1 ตัวละคร ดังรูป 4.26



รูปที่ 4.27 Score

4.2.17 ทดสอบ Run บน Web

1. จุดประสงค์ของการทดลอง

เพื่อให้สามารถเล่นเกมบนเว็บ Browser ได้

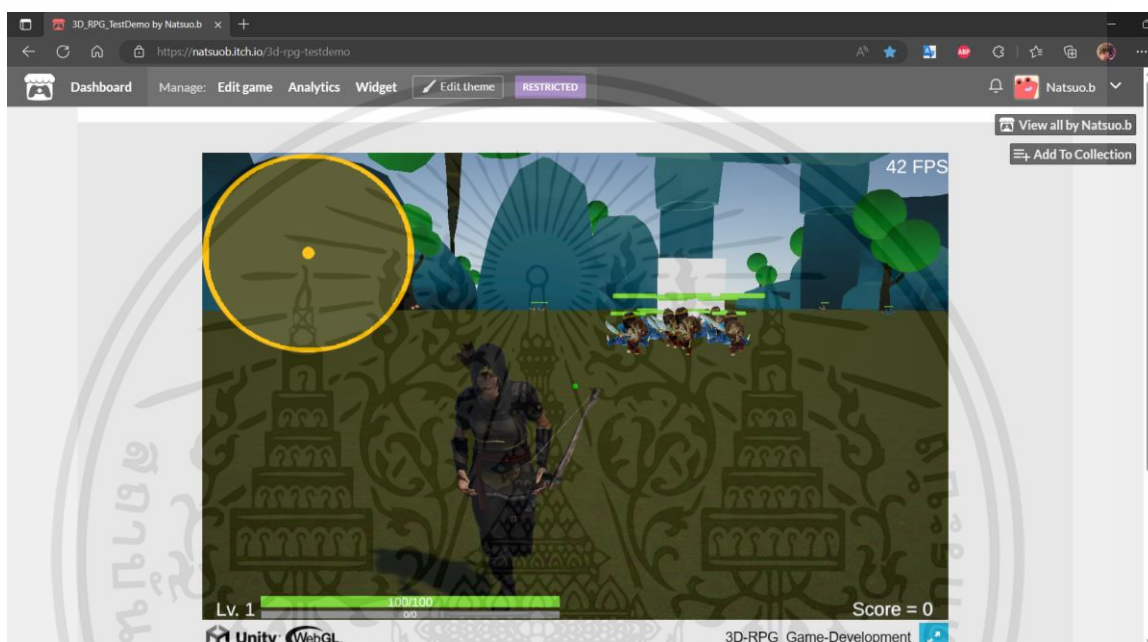
2. วิธีการทดลอง

ทดลอง Upload ไฟล์ Run ตัวเกมขึ้นบนเว็บ ไซต์ itch.io และลองเล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ผลการทดลอง

สามารถเล่นบนเว็บไซต์ได้ ทั้งการเคลื่อนที่ การโจมตี การกระโดด ปุ่มต่างๆ และระบบภายในเกมสามารถทำงานได้ปกติ ยกเว้นห้อง Boss สามารถเล่นได้เฉพาะบนเครื่อง Windows เท่านั้นเพราะระบบ RL ยังไม่รองรับการใช้งานเล่นบนเว็บไซต์ได้ดังรูป 4.27



รูปที่ 4.28 ทดสอบ Run บน Web

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผล อภิปราย และข้อเสนอแนะ

ในการทำโครงการพัฒนาการสร้างและออกแบบเกม สามารถบรรลุวัตถุประสงค์ตามที่ผู้จัดทำได้วางแผนเอาไว้ ซึ่งเกม สามารถทำงานได้ตรงตามที่ต้องการและการประสบปัญหาต่างๆ ในที่นี้ทางผู้จัดทำได้ประสบปัญหาในการปฏิบัติงานหลายๆ อย่างจากโครงการมีข้อเสนอแนะที่จะนำมาปรับปรุงแก้ไขในส่วนต่างๆ ของเกม

5.1 สิ่งที่ทำไปแล้ว

ในการดำเนินโครงการในการพัฒนาการสร้างเกมและออกแบบเกม เริ่มจากการเสนอโครงการต่อคณะกรรมการพิจารณา ทางคณะกรรมการพิจารณาได้เสนอแนะในส่วนต่างๆ แบบผู้จัดทำได้ทำตามข้อเสนอของคณะกรรมการจนได้รับการอนุมัติการทำโครงการแล้วทางผู้จัดทำได้ศึกษาข้อมูลรายละเอียดต่างๆ เกี่ยวกับการพัฒนาการสร้างเกมและออกแบบเกมโดยได้ทำการออกแบบและดำเนินการตามที่วางแผนไว้สำเร็จลุล่วง

ผลการดำเนินโครงการสร้างและออกแบบเกมด้วยโปรแกรมที่ใช้พัฒนาทำให้เราได้ตัวเกมที่มีประสิทธิภาพส่งผลให้ผู้เล่นเกิดความเพลิดเพลินในการเล่นเกมนมากขึ้น ทั้งนี้สิ่งที่ทางผู้จัดทำได้ทำไปแล้วและมีการทดลองไปแล้วมีดังนี้

1. การเคลื่อนที่ของตัวละคร สามารถเคลื่อนที่ได้ทั้งเดินหน้า ถอยหลัง เดินซ้าย เดินขวา วิ่ง และกระโดด เป็นไปตามวัตถุประสงค์ของทางผู้จัดทำได้กำหนดไว้ให้เคลื่อนที่บนพื้นที่และขนาดที่กำหนดเพื่อกันการตกแมพ

2. การโจมตีของตัวละคร

- โจมตีระยะไกล ทดสอบและสามารถโจมตีได้ในระยะไกลสามารถสร้างดาเมจได้ โจมตีผ่านการคลิกเมาส์ซ้าย 1 ครั้ง/การโจมตี

- โจมตีระยะใกล้

3. การโจมตีของ Enemy

- การโจมตีของ Enemy ระดับ Easy สามารถเคลื่อนที่และโจมตีผ่านการสั่งของ AI ที่เขียนขึ้นมาเพื่อใช้ควบคุมได้

- การโจมตีของ Enemy ระดับ Normal สามารถเคลื่อนที่และโจมตีผ่านการสั่งของ AI ที่เขียนขึ้นมาเพื่อใช้ควบคุมได้

- การโจมตีของ Enemy ระดับ Hard สามารถเคลื่อนที่และโจมตีผ่านการสั่งของ AI ที่เขียนขึ้นมาเพื่อใช้ควบคุมได้ ในส่วนของบอสจะใช้ ML เพื่อให้บอสสามารถเรียนรู้และฉลาดได้ด้วยตัวเอง แต่จะไม่สามารถฉลาดเกินมนุษย์ได้

4. Map ของเกม จะแบ่งออกเป็น 2 ส่วนคือ

- ฉากที่ 1 ที่เจอ Enemy ระดับ Easy และ Normal และมีไอเทมครอบเพื่อเก็บกุญแจให้ครบและเข้าไปหาบอส

- ฉากที่ 2 จะเป็นในส่วนของฉากที่ต้องเจอกับบอสเพื่อต่อสู้และจบเกม

5. นำ Machine Learning มาใช้ในการพัฒนาเกม เพื่อให้บอสได้เรียนรู้การต่อสู้และหลบหลีกเอง เพื่อให้เกมไม่ดูน่าเบื่อเพราะมีบอสที่เอาชนะง่าย วิธีนี้เป็นการกระตุ้นผู้เล่นให้รู้สึกอยากเอาชนะกับเกมและมีความท้าทายมากขึ้น

5.2 ปัญหาและอุปสรรค

การดำเนินการของโครงการพัฒนาการสร้างละออกแบบเกมนั้นทางผู้จัดทำได้ประสบปัญหาการดำเนินโครงการหลายอย่างดังต่อไปนี้

5.2.1 มีปัญหาช่วงพัฒนาการ Train Boss โดยใช้ Reinforcement Learning ใน Unity ผ่าน Command ไม่สามารถเก็บค่าที่ Train ได้เพราะตอนกด Run ไฟล์ ไม่ถูกสร้างขึ้นในโพลเดอร์

5.2.2 มีปัญหาตอนรวมโปรเจคเพื่อทำการอัปโหลดงานขึ้นสู่ระบบเพื่อให้เกมสามารถเล่นได้

5.3 ประโยชน์ที่คาดว่าจะได้รับ

5.3.1 ให้ผู้เล่นเกิดความเพลิดเพลิน และผ่อนคลายจากการเล่นเกม

5.3.2 ให้ผู้เล่นได้ประสบการณ์จากการเล่นเกมที่แตกต่างจากเกมทั่วไป

5.4 การอภิปรายผล

ในการสร้างเกมโครงการนี้ได้ใช้โปรแกรม Unity Engine ในการพัฒนาเรื่องการควบคุมตัวละคร และระบบต่างๆ ร่วมกับ Visual Studio 2019 ทำให้ตัวเกมของเรามีประสิทธิภาพ รวมถึงการนำ AI เข้ามาใช้งานกับตัวเกม



ภาคผนวก ก

สิ่งที่ได้ดำเนินการไปแล้ว

ส่วนที่ได้ดำเนินงานไปแล้ว

ในส่วนของระบบที่ได้ดำเนินงานไปแล้ว มีปริมาณงานชิ้นต่ำตามที่กำหนดโดยอ้างอิงจากแผนดำเนินงาน ซึ่งประกอบไปด้วยระบบต่าง ๆ ดังนี้

ข้อ	ระบบ	ผลการดำเนินงาน	ทดสอบ (5)
1	การเดินทาง หน้า-หลัง-ซ้าย-ขวา กระโดดและมูมกลิ้ง	<input checked="" type="checkbox"/>	5
2	การโจมตีระยะใกล้ – ระยะไกล	<input checked="" type="checkbox"/>	5
3	ศัตรูและการโจมตี	<input checked="" type="checkbox"/>	5
4	การใช้ Skill โจมตีของ Boss	<input checked="" type="checkbox"/>	5
5	AI ที่ใช้กับ Enemy	<input checked="" type="checkbox"/>	5
6	EXP ระบบเพิ่ม Level	<input checked="" type="checkbox"/>	5
7	พลังชีวิต HP	<input checked="" type="checkbox"/>	5
8	การตาย Die	<input checked="" type="checkbox"/>	5
9	Mode Easy, Medium, Hard	<input checked="" type="checkbox"/>	5
10	วาร์ป Warp	<input checked="" type="checkbox"/>	5
11	การทำ Damage	<input checked="" type="checkbox"/>	5
12	Action การเคลื่อนไหวตัวละครและ Action อื่นๆ	<input checked="" type="checkbox"/>	5
13	ทดสอบ Run บน Web	<input checked="" type="checkbox"/>	4
14	UI ต่างๆ ภายในเกม	<input checked="" type="checkbox"/>	5
15	Reinforcement Learning (RL).	<input checked="" type="checkbox"/>	4
16	Minimap	<input checked="" type="checkbox"/>	5
17	Score	<input checked="" type="checkbox"/>	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

อินเทอร์เน็ต - วิกีพีเดีย.[online].

Available : <https://th.wikipedia.org/wiki/อินเทอร์เน็ต>

เกม - วิกีพีเดีย.[online].

Available : <https://th.wikipedia.org/wiki/เกม>

เกมออนไลน์ - วิกีพีเดีย.[online].

Available : <https://th.wikipedia.org/wiki/เกมออนไลน์>

ทฤษฎีเกม - วิกีพีเดีย.[online].

Available : <https://th.wikipedia.org/wiki/ทฤษฎีเกม>

Algorithm.[online].

Available : <https://en.wikipedia.org/wiki/Algorithm>

A* Algorithm.[online].

Available : <https://brilliant.org/wiki/a-star-search/>

A* search algorithm.[online].

Available : https://en.wikipedia.org/wiki/A*_search_algorithm

Reinforcement Learning (RL).[online].

Available : <https://tupleblog.github.io/reinforcement-learning/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

npcs-with-machine-learning.[online].

Available : <https://community.arm.com/arm-community-blogs/b/graphics-gaming-and-vr-blog/posts/intelligent-npcs-with-machine-learning>

AI.[online].

Available : <https://www.aiforall.or.th/>

NPC's in games with ML Agents in Unity.[online].

Available : <https://community.arm.com/arm-community-blogs/b/graphics-gaming-and-vr-blog/posts/intelligent-npcs-with-machine-learning>

Proximal Policy Optimization (PPO) .[online].

Available : <https://openai.com/research/openai-baselines-ppo>

Generative Adversarial Imitation Learning (GAIL) .[online].

Available : <https://arxiv.org/abs/1606.03476>