



รายงานโครงการวิทยาการคอมพิวเตอร์

การพัฒนาเกมต่อสู้เอาชีวิตรอดโดยใช้ Unreal Engine

Development of Survival RPG Game using Unreal Engine

ผู้จัดทำ

นายวิศรุต หอมแก่นจันทร์ รหัสประจำตัวนิสิต 6310401165

อาจารย์ที่ปรึกษา

ดร.พิศุทธิ์ วิเศษสิงห์
ผศ.ดร.อรุณรัณ อิมสมบัติ

ภาควิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
ภาคปลาย ปีการศึกษา 2566

โครงการวิทยาการคอมพิวเตอร์

ชื่อเรื่อง การพัฒนาเกมต่อสู้ เอาชีวิตรอด โดยใช้ Unreal Engine

Development of Survival RPG Game using Unreal Engine

ผู้จัดทำ นายวิศรุต ห้อมแก่นจันทร์ รหัสประจำตัวนิสิต 6310401165

อาจารย์ที่ปรึกษาหลัก

(ดร. พิศุทธิ์ วิเศษสิงห์)

...../...../.....

อาจารย์ที่ปรึกษาร่วม

(ผศ. ดร. อรุณรัณ อิ่มสมบัติ)

...../...../.....

หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์

(ผศ. ดร. ผกาเกษตร วัตถุยา)

...../...../.....

นายวิศรุต หอมแก่น จันทร์. 2566. การพัฒนาเกมต่อสู้อาชีวิตรอดโดยใช้ Unreal Engine. โครงการวิทยาการคอมพิวเตอร์ ปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.

อาจารย์ที่ปรึกษา:

ดร. พิศุทธิ์ วิเศษสิงห์

บทคัดย่อ

ในปัจจุบันเกมได้เข้ามามีบทบาทในชีวิตประจำของผู้คน เนื่องจากเป็นสื่อที่ให้ความบันเทิง และ สามารถนำมาเป็นสื่อการสอนหรือฝึกฝนให้ผู้เล่นได้มีความอดทน และความพยายามผ่านด่านไปสู่จุดจบของเกมด้วยตนเอง ด้วยการทำให้เกมมีความท้าทาย และทำให้ผู้เล่นต้องใช้ทักษะการคิด การวางแผน

ดังนั้นผู้จัดทำจึงได้พัฒนาเกมต่อสู้อาชีวิตรอดจากศัตรูประเภทต่างๆ โดยเกมนี้มีมุ่งมองการเล่นแบบ Top-Down View มีความท้าทาย และให้ความตื่นเต้นจากการสุ่มการเกิดของศัตรูและการสุ่มพัฒนาความสามารถของตัวละครผู้เล่น ทุกครั้งที่ผู้เล่นเริ่มเกมใหม่ การพัฒนาศัตรูให้มีความสามารถในการตอบสนองกับผู้เล่นด้วยระบบ Behavior Tree และ AI Perception ของ Unreal Engine

เกมต่อสู้อาชีวิตรอดนี้มีจุดมุ่งหมายให้ผู้เล่นสามารถอาชีวิตรอดให้ถึงเวลาที่กำหนด โดยการกำจัดศัตรูเพื่อได้รับค่าประสบการณ์ นำไปใช้ในการพัฒนาตัวละครให้สามารถต่อสู้กับศัตรูจนจบเกมได้ โดยซอฟต์แวร์ที่ใช้ในการพัฒนาเกมนี้คือ Unreal Engine 5 ซึ่งเป็นเครื่องมือที่ได้รับความนิยมและช่วยอำนวยความสะดวกในการพัฒนาเกม

Witsarut Homkanchan. 2023. **Development of Survival RPG Game using Unreal Engine**. Computer Science Project. Bachelor of Science in Computer Science, Department of Computer Science, Faculty of Science, Kaset-sart University.

Project Advisor:

Prof. Pisut Wisessing

Abstract

Nowaday, games have taken on a vital role in people's daily lives, as mediums that provide entertainment and can serve as tools for teaching or training players to cultivate patience and perseverance, guiding them through challenges toward the final scenes under their own power. By injecting games with difficulty and demanding players employ their thinking and planning skills, developers have crafted combat survival games pitting players against myriad foes.

This particular game employs a top-down perspective, challenging and exciting players through the random spawning of enemies and randomized character ability progression each new playthrough. Foes are imbued with reactive capabilities thanks to behavior trees and the AI perception of Unreal Engine.

The aim is for players to survive until the designated time by eliminating enemies to accrue experience points for further character upgrades, allowing them to battle through to the culmination. This game was crafted using Unreal Engine 5, a popular and facilitating tool in game development.

กิติกรรมประกาศ

โครงการนิวัติกรรมคอมพิวเตอร์เรื่อง การพัฒนาเกมต่อสู้ เอาชีวิตรอด โดยใช้ Unreal Engine ฉบับนี้สำเร็จลุล่วงด้วยดีเนื่องจากได้รับคำแนะนำและเสนอแนวทางแก้ไข รวมทั้งการติดตามการจัดทำโครงการ จากอาจารย์ที่ปรึกษาประจำโครงการ ดร.พิศุทธิ์ วิเศษสิงห์ และ ผศ.ดร.อรุณรัตน อิ่มสมบติ ที่ได้ให้คำแนะนำ ให้คำปรึกษา เสนอแนวทางแก้ไข รวมถึงติดตามการจัดทำโครงการตลอดระยะเวลาการทำ

ผู้จัดทำโครงการของขอบคุณทุกท่านที่มีส่วนร่วมในการให้ความรู้และคำปรึกษาต่างๆ ที่เกี่ยวข้องกับโครงการนี้ จนสำเร็จลุล่วงในที่สุด

ผู้จัดทำ
นายวิศรุต หอมแก่น จันทร์ 6310401165

สารบัญ

1	บทนำ	8
1.1	ที่มาและความสำคัญ	8
1.2	วัตถุประสงค์	8
1.3	ขอบเขต	8
1.4	ประโยชน์ที่คาดว่าจะได้รับ	9
1.5	แผนการดำเนินงาน	9
2	งานที่เกี่ยวข้อง	10
2.1	เทคโนโลยีที่เกี่ยวข้อง	10
2.1.1	Unreal Engine	10
2.1.2	Blender 3D	10
2.1.3	Mixamo	11
2.1.4	Github	11
2.2	แนวคิดทฤษฎีที่เกี่ยวข้อง	11
2.2.1	หลักการออกแบบเกม	11
2.2.2	การสร้างเกม	12
3	การวิเคราะห์และออกแบบระบบ	14
3.1	การวิเคราะห์และออกแบบเกม	14
3.1.1	แนวคิดการออกแบบระบบเกม	14
3.1.2	ตัวละครของผู้เล่น	14
3.1.3	การควบคุมตัวละคร	15
3.1.4	ศั不住	15
3.1.5	ความสามารถของตัวละคร	18
3.1.6	ไอเทม	20
3.2	การออกแบบส่วนต่อประสานกับผู้ใช้	20
3.2.1	หน้าเมนูหลัก	21
3.2.2	หน้าเล่นเกม	21
3.2.3	หน้าสรุปผลการเล่น	22
4	การพัฒนาระบบ	23
4.1	การพัฒนาแผนที่ของเกม	23
4.2	การพัฒนาตัวละครผู้เล่น	24
4.3	การพัฒนาศั不住	26

4.4	ระบบเกมเพย์	27
4.4.1	ระบบการเกิดของศัตรู	27
4.4.2	ระบบการอัพเกรดตัวละคร	28
5	การทดสอบ ผลลัพธ์ และการประเมิน	29
5.1	ระบบตัวละครผู้เล่น	29
5.2	ระบบศัตรู	29
5.3	ระบบสิ่งแวดล้อม	30
6	บทสรุป ปัญหา และข้อเสนอแนะ	31
6.1	บทสรุป	31
6.2	ปัญหาและข้อเสนอแนะ	31
6.3	แนวทางในการพัฒนาต่อไป	31

สารบัญรูปภาพ

1.1	ตารางแสดงการดำเนินงาน	9
3.1	รูปตัวละครผู้เล่น	14
3.2	ตารางแสดงปุ่มการควบคุมตัวละครของผู้เล่น	15
3.3	รูปศัตรู Zombie	15
3.4	รูปศัตรู Ninja	16
3.5	รูปศัตรู Witch	16
3.6	รูปศัตรู Giant Zombie	17
3.7	รูปศัตรู Witch Goblin	17
3.8	รูป Active Skill: Ice bolt	18
3.9	รูป Active Skill: Fire ball	18
3.10	รูป Active Skill: Wind Blade	19
3.11	รูป Passive Skill: Increase Max Health	19
3.12	รูป Passive Skill: Increase Movement Speed	19
3.13	รูป Passive Skill: Increase Ability Damage	20
3.14	รูป Passive Skill: Cooldown Reduction	20
3.15	รูปหน้า Main Menu	21
3.16	รูปหน้าจอเล่นเกม	21
3.17	รูปหน้าสรุปผลการเล่น	22
4.1	แผนที่ของเกม	23
4.2	หน้าสร้างตัวละคร	24
4.3	Blueprint การบังคับตัวละคร	24
4.4	Blueprint ความสามารถของตัวละคร	25
4.5	หน้าเว็บไซต์ Mixamo	26
4.6	Animation Blueprint	26
4.7	Behavior Tree	27
4.8	Data Table ข้อมูลศัตรู	27
4.9	Blueprint การเกิดของศัตรู	28
4.10	หน้าจอ Level Up	28
5.1	ตารางแสดงการทดสอบการทำงานของตัวละครผู้เล่น	29
5.2	ตารางแสดงการทดสอบการทำงานของศัตรู	29
5.3	ตารางแสดงการทดสอบการทำงานของสิ่งแวดล้อม	30

บทที่ 1

บทนำ

1.1. ที่มาและความสำคัญ

เกม เป็นสื่อให้ความบันเทิงที่สามารถเข้าถึงได้ง่าย และได้รับความนิยมเป็นอย่างมาก เนื่องจากผู้บริโภคสามารถเข้าไปมีส่วนร่วมกับเกมได้ ทำให้ผู้เล่นได้ใช้กระบวนการคิด แก้ปัญหา และได้รับความสนุกสนานไปพร้อมๆ กันได้ โดยผู้จัดทำโครงงานมีความชื่นชอบในการเล่นเกม

ผู้จัดทำจึงมีความสนใจในเรื่องของการพัฒนาเกม ทำให้ผู้จัดทำต้องการจะศึกษาหาความรู้ในขั้นตอนการพัฒนาเกม ศึกษาวิธีการใช้งานเครื่องมือต่างๆ ซึ่งจะสามารถเพิ่มทักษะและความรู้ให้กับผู้จัดทำ เพื่อสามารถที่จะนำความรู้ที่ได้จากการทำโครงงานไปต่อยอดในการทำงานในอนาคตได้

จึงโปรแกรมที่ผู้จัดทำเลือกใช้ในการพัฒนาโครงงานนี้คือ Unreal Engine 5 ซึ่งเป็นเครื่องมือที่ได้รับความนิยมเป็นอย่างมาก เนื่องจาก Unreal Engine มีความสามารถในการสร้างกราฟิกที่สวยงามและสมจริงในระดับสูง และ มีเครื่องมือพัฒนาแบบ Visual Scripting ที่เรียกว่าระบบ Blueprint ที่ช่วยให้นักพัฒนาสามารถสร้างโลジิกและระบบต่างๆ ได้อย่างง่ายดายและรวดเร็ว โดยไม่จำเป็นต้องเขียนโค้ดมากนัก

ดังนั้นผู้จัดทำจึงมีความสนใจที่จะสร้างเกมต่อสู้ เอาชีวิตรอด ที่จะทำให้ผู้เล่นได้รับความเพลิดเพลิน สนุกสนาน และเกิดความท้าทายระหว่างการเล่นเกม ผู้เล่นจะต้องคิดวางแผนการพัฒนาตัวละครเพื่อที่จะสามารถเอาชีวิตรอดจากศัตรุประเภทต่างๆ ได้จนจบเกม โดยการพัฒนาความสามารถของผู้เล่นจะเกิดจากการสูม ทำให้เกมมีความน่าสนใจและไม่น่าเบื่อ และอีกสิ่งที่สำคัญที่สุดในการเล่นเกมคือ ศัตรุ ซึ่งศัตรุในเกมนี้จะมีหลายประเภทและมีความสามารถที่แตกต่างกัน โดยจะใช้ระบบ Behavior Tree และ AI Perception ในการควบคุมพฤติกรรมของศัตรุ

1.2. วัตถุประสงค์

- เพื่อออกแบบเกมที่ทำให้ผู้เล่นได้รับความสนุก
- เพื่อให้ผู้เล่นได้คิดวิเคราะห์ การพัฒนาตัวละคร
- เพื่อพัฒนาเกมที่ศัตรุสามารถตอบสนองกับผู้เล่นได้

1.3. ขอบเขต

1. กีฬากับเกม

เป็นเกมเอาชีวิตรอด ที่ต้องบังคับตัวละครผ่านมุมมองการเล่นแบบ Top-Down View โดยจะใช้ปุ่ม WASD ในการเคลื่อนที่ และ ใช้เมาส์ในการกำหนดทิศทางการโจมตี

2. ตัวละครผู้เล่น

ผู้เล่นสามารถพัฒนาตัวละครจากการเก็บค่าประสบการณ์ที่ได้จากการกำจัดศัตรุ โดยเมื่อเก็บค่าประสบการณ์ได้ถึงกำหนด ผู้เล่นจะได้เลือกอัพเกรดความสามารถต่างๆ ที่เกิดจากการสูมของระบบ

3. ศัตรู

จะมีศัตรูรรบด้วยทั้งหมด 3 ประเภท และมีบอสอีก 2 ประเภท ได้แก่

- Zombie จะทำการเดินเข้ามาหาผู้เล่นเพื่อทำการโจมตี
- Ninja จะวิ่งเข้ามาหาผู้เล่นด้วยความเร็วสูงเพื่อโจมตี และมีความสามารถในการหลบการโจมตีของผู้เล่นได้
- Witch จะมีความสามารถในการโจมตีระยะใกล้ และสามารถหาตำแหน่งการยืนที่เหมาะสมให้กับตัวเองได้
- Giant Zombie จะมีพลังชีวิตมาก และ โจมตีได้รุนแรง
- Witch Goblin จะโจมตีจากระยะไกล และสามารถโจมตีต่อเนื่องได้หลายรอบในเวลาสั้น

4. เป้าหมายของเกม

ผู้เล่นจะต้องเอาชีวิตรอดจากศัตรุหลากหลายประเภท หากผู้เล่นเอาชีวิตรอดได้ครบเวลาที่กำหนด ก็จะถือว่าผู้เล่นชนะเกม

1.4. ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้เล่นได้รับความสนุกสนานจากการเล่นเกม
2. ผู้เล่นได้ใช้ทักษะการคิด วิเคราะห์ ในการเล่นเกม

1.5. แผนการดำเนินงาน

ในรูปที่ 1.1 จะเป็นตารางแสดงถึงแผนการดำเนินงาน

ขั้นตอนการทำงาน	ระยะเวลา พ.ศ. 2566 -2567				
	พฤษภาคม	มิถุนายน	กรกฎาคม	กันยายน	ตุลาคม
กำหนดและออกแบบขอบเขตโครงการ					
ศึกษาข้อมูลและเครื่องมือที่ใช้					
พัฒนาระบบต่างๆ					
ทดสอบและแก้ไข					
จัดทำรายงาน					

รูปที่ 1.1: ตารางแสดงการดำเนินงาน

บทที่ 2

งานที่เกี่ยวข้อง

เนื้อหาในบทที่ 2 เป็นส่วนที่อธิบายเกี่ยวกับทฤษฎีที่เกี่ยวข้องที่นำมาใช้ในโครงการเพื่อให้มีความเข้าใจในหลักการ และเทคโนโลยีที่นำมาใช้ในการทำโครงการ

ในโครงการฉบับนี้ได้พัฒนาเกมต่อสู้ เอาชีวิตรอด แบบ Top-Down View โดยใช้โปรแกรม Unreal Engine ซึ่งเป็นโปรแกรมที่ช่วยอำนวยความสะดวกในการสร้างเกม โดยใช้ระบบ Visual Scripting ที่เรียกว่าระบบ Blueprint ในการเขียน Script ที่ใช้ในการสร้างเกม โดยในบทนี้จะกล่าวถึงเทคโนโลยีที่เกี่ยวข้อง และเทคโนโลยีที่ใช้ ดังนี้

2.1 เทคโนโลยีที่เกี่ยวข้อง

- 2.1.1 Unreal Engine
- 2.1.2 Blender 3D
- 2.1.3 Mixamo
- 2.1.4 GitHub

2.2 แนวคิดทฤษฎีที่เกี่ยวข้อง

- 2.2.1 หลักการออกแบบเกม
- 2.2.2 การสร้างเกม

2.1. เทคโนโลยีที่เกี่ยวข้อง

2.1.1. Unreal Engine

Unreal Engine เป็นเครื่องมือในการพัฒนาเกมและแอปพลิเคชัน 3D ที่มีความสามารถที่หลากหลาย สามารถใช้สำหรับสร้างเกม 3D, แอปพลิเคชัน VR/AR, ภาพนิ่ง 3D, และโมเดลสามมิติ ที่พัฒนาโดย Epic Games และมีประวัติยาวนานในการใช้งานในอุตสาหกรรม

มีหลักการทำงานสำคัญคือ ใช้กราฟิกสามมิติ (3D) และเทคโนโลยีเกมเพื่อสร้างโลกเสมือน และมีระบบการเขียนสคริปต์ (Blueprints) ที่ใช้งานง่ายและมีความสามารถในการกำหนดการควบคุมต่างๆ อีกทั้งยังมีระบบแสงและเงา ช่วยให้ภาพมีความสมจริง และมีระบบพิสิกส์ที่สมจริงในการจำลองการเคลื่อนไหวและการกระทำต่างๆ

2.1.2. Blender 3D

Blender 3D เป็นซอฟต์แวร์เสรี สำหรับงานคอมพิวเตอร์กราฟิกสามมิติ สามารถใช้สร้างโมเดลสามมิติ, จัดการการเคลื่อนไหวของตัวละครโดยใช้กระดูก เป็นต้น

Blender เป็นโปรแกรมที่มีขนาดเล็ก สามารถทำงานได้บนระบบปฏิบัติการหลายรูปแบบ มีความสามารถในการทำงานทั่วไป และโมเดล ได้ใกล้เคียงหรือแม้แต่สูงกว่าโปรแกรมสามมิติระดับสูงอื่นๆ ในหลายกรณี เช่น สามารถรองรับโครงสร้างพื้นฐานเรขาคณิต

ได้หลายอย่าง, รองรับการนำเข้าไฟล์จากโปรแกรมอื่นๆ, มีเครื่องมือสำหรับการทำแอนิเมชัน, มีเครื่องมือสำหรับใช้ตัดต่อและตกแต่งวีดีโอ, สามารถเขียนโปรแกรมเสริมการทำงานได้ด้วยภาษา ไฟทอนสคริป เป็นต้น

2.1.3. Mixamo

Mixamo เป็นเว็บแพลตฟอร์มที่สามารถสร้างและแก้ไขตัวละคร 3 มิติ (3D characters) ได้อย่างง่าย โดยมีคุณสมบัติที่เป็นจุดเด่นดังนี้

1. Mixamo ช่วยให้สามารถสร้างตัวละคร 3 มิติด้วยตัวเอง ๆ โดยใช้ตัวละครที่มีอยู่ในคลังข้อมูลของ Mixamo หรืออัปโหลดตัวละครที่ผู้ใช้สร้างขึ้นมาเอง และ Mixamo จะทำการใส่กระดูก (skeleton) ให้ตัวละครให้พร้อมใช้งานในการสร้างแอนิเมชันต่อไปได้
2. สามารถนำแอนิเมชันที่มีอยู่ในคลังข้อมูลของ Mixamo มาใส่ในตัวละครของผู้ใช้ และแก้ไขหรือปรับแต่งการเคลื่อนไหวตามต้องการ ทำให้ควบคุมลักษณะการเคลื่อนไหวของตัวละครได้
3. สามารถดาวน์โหลดตัวละครและแอนิเมชันในรูปแบบต่าง ๆ พร้อมใช้ในงานกราฟิกเคลื่อนไหว วีดีโอกลีฟ หรืองานภาพวิดีโอ

Mixamo จึงเป็นเครื่องมือที่มีประโยชน์สำหรับนักออกแบบ นักพัฒนาเกม ศิลปิน และผู้สนใจในการสร้างตัวละคร 3 มิติ

2.1.4. Github

Github คือ website Git (version control repository) ที่อยู่บนอินเตอร์เน็ต มีการทำงานแบบ เดียวกับ Git แต่สามารถเข้าถึงข้อมูลและจัดการผ่านเว็บโดยไม่ต้องเสียค่าใช้จ่าย หรือลงทุนตั้ง server เพื่อติดตั้ง Git แต่ code project ทั้งหมดจะถูกทำให้คนอื่น ๆ สามารถเห็นได้ด้วย โดยประโยชน์ของ Github คือ สามารถตรวจสอบเวอร์ชันย้อนหลังของ source code ได้และช่วยในการพัฒนาซอฟต์แวร์แบบเป็นทีม

2.2. แนวคิดทฤษฎีที่เกี่ยวข้อง

2.2.1. หลักการออกแบบเกม

หลักการออกแบบเกมนั้นไม่มีกฎเกณฑ์ตายตัว แต่มีปัจจัยที่มีความเกี่ยวข้องกับความรู้สึกของ ผู้เล่นเกมโดยตรง เรายังเริ่มจากสมมุติตัวเองเป็นผู้เล่น ลองคิดดูว่าผู้เล่นอย่างไได้อะไรจากเกม ในเกม ของเรานั้น มีส่วนที่ไม่จำเป็นหรือมีส่วนที่ต้องเพิ่มเติม หรือไม่ ซึ่งเป็นกระบวนการที่ต้องอาศัย จินตนาการ และความคิดสร้างสรรค์โดยแบ่งออกเป็นขั้นตอนดังนี้

- **Interactive Design**

การออกแบบระบบควบคุมในเกม เช่น ระบบควบคุมตัวละคร เป็นต้น โดยการออกแบบ ต้องคำนึงถึงว่าเราต้องให้เล่นควบคุมเกมให้มีประสิทธิภาพมากที่สุด การควบคุมเกม

- **Graphic Design**

เกมที่มีกราฟิกสวยงามเป็นสิ่งที่ดึงดูดผู้เล่นได้เป็นอย่างดี องค์ประกอบของกราฟิกที่ดีนั้น ส่วนใหญ่มาจากหลักการออกแบบพื้นฐาน เช่น ทฤษฎีสี จังหวะ การจัดวาง ความกลมกลืน ๆ ฯลฯ รวมถึงลักษณะ มุ่งมองที่เหมาะสมกับเกม สำหรับเกมที่มีเนื้อเรื่อง การออกแบบตัวละครต้องมีความพิถีพิถันเป็นอย่างมาก

- **Game Play Design**

คือลักษณะการเล่นหลักของเกมนั้น ๆ รวมไปถึงกฎข้อบังคับภายในเกม เช่น การเดินของตัวละคร การเดินของสัตว์ประหลาด การช่วยเหลือเพื่อน การเลือกตัวเลือกต่างๆ การเปลี่ยนจักระหว่าง ห้อง เกมเพลย์เป็นส่วนสำคัญต่อความสนุกของเกม การออกแบบเกมจึงควรให้มีความสำคัญกับปัจจัย นี้เป็นอันดับแรก

- Concentration

เกมจะต้องดึงดูดและผู้เล่นสามารถอยู่กับเกมได้นาน

1. เกมควรจะมีสิ่งกระตุ้นอย่างหลากหลายในเกม เพื่อให้ผู้เล่นได้จดจ่อและ เพลิดเพลินไปกับสิ่งเหล่านั้น เช่น มีตัวเลือกในเกมอย่างหลากหลายให้เลือกใช้ ซึ่งตัวเลือกแต่ ละอย่างจะให้ผลที่แตกต่างกันออกไป เป็นต้น
2. สิ่งกระตุ้นภายในเกมจะต้องมีความน่าสนใจเพียงพอ ซึ่งจะแตกต่างกันไปตาม ลักษณะพฤติกรรมของผู้เล่น
3. เกมควรจะเป็นที่น่าสนใจของผู้เล่นอย่างรวดเร็ว และรักษา rate ดับความน่าสนใจอัน นี้ให้ได้ทั้งเกม
4. เกมแต่ละเกมจะมีเป้าหมายหลัก ผู้เล่นไม่ควรจะทำสิ่งที่ไม่ใช่เป้าหมายของหลัก เกมซึ่งจะทำให้เกมน่าเบื่อ

- Challenge

เกมควรมีการท้าทายอย่างเหมาะสมสมกับระดับทักษะของผู้เล่นตามกลุ่มเป้าหมาย

1. ความท้าทายในเกมจะต้องเหมาะสมสมกับระดับทักษะของผู้เล่น ไม่ควรยากหรือง่าย จนเกินไป
2. ระดับความท้าทายควรเพิ่มขึ้นตามทักษะการเล่นที่พัฒนาขึ้นของผู้เล่นตลอดทั้ง เกม เช่น เกมในช่วงแรกจะง่ายก่อน เพื่อเปิดโอกาสให้ผู้เล่นได้พัฒนาทักษะการเล่น และจะ ยกขึ้นไปเรื่อยๆ ในช่วงต่อๆ ไป
3. เกมควรมีความท้าทายใหม่ๆ เมื่อถึงจังหวะเวลาที่เหมาะสม ความท้าทายในเกม ควรมีหลากหลายเพื่อลดความน่าเบื่อ สร้างความแปลกใหม่ และความคาดไม่ถึงให้กับผู้เล่น

- Control

ควรทำให้ผู้เล่นสามารถควบคุมสิ่งต่างๆ ในเกมได้เป็นอย่างดี

1. ควรทำให้ผู้เล่นสามารถควบคุมตัวละครได้เป็นอย่างดี ไม่ติดขัด
2. ควรทำให้ผู้เล่นสามารถควบคุม Interactive Device ได้อย่างสะดวก
3. ควรทำให้ผู้เล่นสามารถควบคุมการเริ่มเกม การหยุดเกมได้อย่างสะดวก
4. เมื่อมีความผิดพลาดร้ายแรงในเกม ควรทำให้ผู้เล่นได้รับการช่วยเหลือเพื่อแก้ไข ข้อผิดพลาด
5. เกมควรทำให้ผู้เล่นควบคุมการกระทำของตัวเอง ซึ่งเป็นอิสระในการเล่นเกมใน แบบที่ผู้เล่นต้องการ

- Clear Goals

ควรให้เกมมีการเคลียร์จุดมุ่งหมายต่าง ๆ ที่ตั้งไว้ในเวลาที่เหมาะสม

- Feedback

ควรให้ผู้เล่นได้รับ “ผลที่ได้รับจากการกระทำ” ที่เหมาะสมในเวลาที่เหมาะสม

- Storytelling

คืออส่วนของเนื้อเรื่องหรือเนื้อหาภายในเกม การสอดแทรกเรื่องราวลงไปในเกมสามารถสร้าง ความสนุกสนานและความบันเทิงให้กับผู้เล่นได้เป็นอย่างดี

2.2.2. การสร้างเกม

จะแบ่งออกเป็น 6 ส่วนในการวางแผนการสร้างเกมจนออกมาเป็นเกมที่สมบูรณ์

- ร่างโครงสร้างคิดคอนเซปของเกมและจุดขายของเกมว่าจะนำเสนออะไรและร่าง เป็นโครงสร้างของมาหากได้รับอนุมัติจะเริ่มขั้นตอนการทำเกม

- กำหนดเด้าโครงเด้าโครงที่ว่ามีคือกำหนดวิธีการสร้างเกมที่จะรวมสิ่งจำเป็น ทุกอย่างที่ต้องกำหนดเพื่อสร้างเกมในอุดมคติ ซึ่งอาจมีการลงละเอียดบุกเบิกอย่างมาก
- ลงมือปฏิบัติงานเป็นการปฏิบัติงานแต่ละฝ่ายตามรายละเอียดที่ได้กำหนดในเด้าโครงซึ่งในระหว่างการทำงานจำเป็นต้องสื่อสารกับแผนกอื่นที่เกี่ยวข้องมาก
- ฉบับ Demo คือ ฉบับที่ใช้ในการทดสอบเกม หรือเรียกได้ว่าเป็นฉบับที่ยังไม่เสร็จ บูรณา เพื่อใช้ตรวจสอบเพื่อแก้ไขจุดบกพร่อง
- แก้ Bug เกมขั้นตอนที่ขาดไม่ได้ในการสร้างเกมเพื่อแก้ไขจุดบกพร่องหลังจากลอง เล่นฉบับ Demo แล้ว
- ฉบับ Master เสร็จสมบูรณ์หลังจากแก้ Bug ในเกมทั้งหมด และตรวจสอบความถูกต้องจากเด้าโครงเกมแล้วนำมาเป็นแนวทางในการวางแผนในการสร้างเกมจนจะออกมาเป็นเกมที่สมบูรณ์โดยจะต้องคำนึงถึงข้อต่างๆ ดังนี้ ร่างโครงร่าง กำหนดเด้าโครง ลงมือทำตามแผน เกมฉบับทดสอบ แก้ไขจุดเสีย ข้อผิดพลาด ของเกมจากเกม ฉบับทดสอบ เกมฉบับสมบูรณ์ พร้อมวางขาย เป็นตามขั้นตอน และ ประเมินผลงานว่าอยู่ในระดับไหน ควรแก้ไขหรือทำยังไงต่อ

บทที่ 3

การวิเคราะห์และออกแบบระบบ

เนื้อหาในบทที่ 3 เป็นส่วนที่อธิบายการวิเคราะห์และออกแบบระบบของเกมที่ผู้จัดทำพัฒนา โดยจะมีการอธิบายถึงระบบ หรือเกมต้นแบบที่ผู้จัดทำไปศึกษามา เพื่อให้เข้าใจถึงข้อดีข้อเสียของเกมต้นแบบ และนำข้อดีข้อเสียที่ได้มาพัฒนา และออกแบบ เป็นเกมที่ผู้จัดทำพัฒนา

3.1. การวิเคราะห์และออกแบบเกม

3.1.1. แนวคิดการออกแบบระบบเกม

ทางผู้จัดทำมีแนวคิดออกแบบการเล่นเกม ต่อสู้อาชีวิตรอดโดยการบังคับตัวละครหลัก ผ่านมุมมองแบบ Top-Down View โดยมีวัตถุประสงค์คือ ผู้เล่นจะต้องต่อสู้กับศัตรูด้วยเวทย์มนตร์ต่างๆ และอาชีวิตรอดให้ครบเวลาที่กำหนด และผู้เล่นจะต้องคิดวิเคราะห์ในการเลือกพัฒนาความสามารถของตัวละครอย่างเหมาะสมให้สามารถเอาชีวิตรอดไปจนจบเกมได้

3.1.2. ตัวละครของผู้เล่น

ตัวละครที่ผู้เล่นจะได้ควบคุมคือ ตัวละครพ่อมด โดยตัวละครสามารถเคลื่อนที่ด้วยการเดิน และสามารถโจมตีโดยใช้พลังเวทย์ มนต์ได้



รูปที่ 3.1: รูปตัวละครผู้เล่น

3.1.3. การควบคุมตัวละคร

จากรูปภาพที่ 3.2 จะเป็นตารางแสดงปุ่มที่ผู้ใช้งานสามารถควบคุมตัวละครของผู้เล่นได้

ปุ่ม	สถานะปุ่ม
w	เคลื่อนที่ไปด้านบน
S	เคลื่อนที่ไปด้านล่าง
A	เคลื่อนที่ไปด้านซ้าย
D	เคลื่อนที่ไปด้านขวา
Mouse Left	ตอบสนองกับปุ่มค้างๆ

รูปที่ 3.2: ตารางแสดงปุ่มการควบคุมตัวละครของผู้เล่น

3.1.4. ศัตรู

ศัตรูในเกมนี้จะมีทั้งหมด 5 รูปแบบ ดังนี้

Zombie

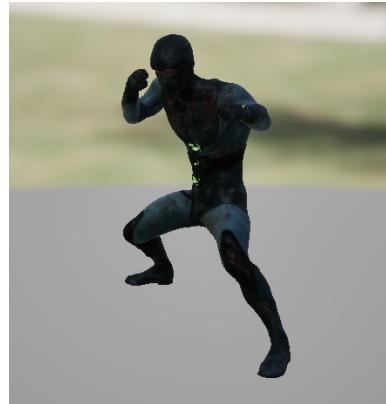
มีรูปร่างดังรูปที่ 3.3 โดย Zombie จะมีลักษณะนิสัยคือ เดินเข้ามาหาผู้เล่นเพื่อทำการโจมตี



รูปที่ 3.3: รูปศัตรู Zombie

Ninja

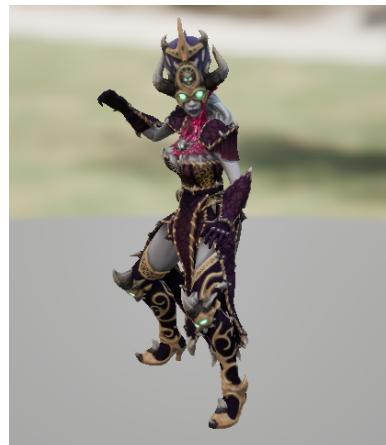
มีรูปร่างดังรูปที่ 3.4 โดย Ninja จะมีลักษณะนิสัยคือ Ninja จะวิ่งเข้ามาหาผู้เล่นเพื่อทำการโจมตี และมีโอกาสที่จะกระโดดหลบการโจมตีของผู้เล่นได้



รูปที่ 3.4: รูปตัว Ninja

Witch

มีรูปร่างดังรูปที่ 3.5 โดย Witch จะมีลักษณะนิสัยคือ เดินเข้ามาบริเวณใกล้ๆ กับผู้เล่น และทำการโจมตีจากระยะใกล้ อีกทั้งยังสามารถเปลี่ยนตำแหน่งของตัวเองให้รักษาระยะห่างจากผู้เล่น



รูปที่ 3.5: รูปตัว Witch

Giant Zombie

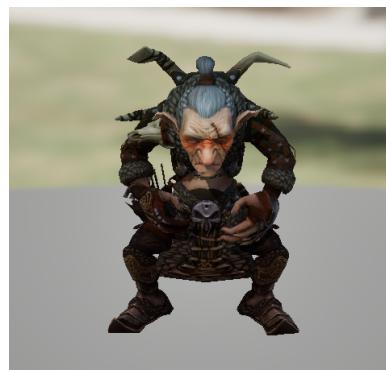
มีรูปร่างดังรูปที่ 3.6 โดย Giant Zombie จะมีลักษณะนิสัยคือ Giant Zombie จะเดินเข้ามาหาผู้เล่นเพื่อทำการโจมตี มีความแตกต่างจาก Zombie ธรรมดาคือมีขนาดใหญ่และมีเลือดมากกว่า



รูปที่ 3.6: รูปศัตรู Giant Zombie

Witch Goblin

รูปร่างดังรูปที่ 3.7 โดย Witch Goblin จะมีลักษณะนิสัยคือ Witch Goblin จะเดินเข้ามาบวิเณ่กล้ากับผู้เล่น และทำการโจมตีจากระยะใกล้ อีกทั้งยังสามารถเปลี่ยนตำแหน่งของตัวเองให้รักษาระยะห่างจากผู้เล่น และ สามารถโจมตีต่อเนื่องได้



รูปที่ 3.7: รูปศัตรู Witch Goblin

3.1.5. ความสามารถของตัวละคร

ความสามารถของตัวละครจะแบ่งเป็น 2 ประเภท ได้แก่ Active Abilities และ Passive Abilities

Ice bolt

เป็นความสามารถเริ่มต้นของตัวละครผู้เล่น จะทำการโจมตืออกไปเป็น Projectile สร้างความเสียหายให้กับศัตรุตัวแรกที่โดน



รูปที่ 3.8: รูป Active Skill: Ice bolt

Fire ball

เป็นการโจมตืออกไปเป็น Projectile และจะระเบิดเมื่อสัมผัสถูกตัวต่างๆ และจะสร้างความเสียหายให้กับศัตรุทุกตัวที่อยู่ในรัศมีการระเบิด



รูปที่ 3.9: รูป Active Skill: Fire ball

Wind Blade

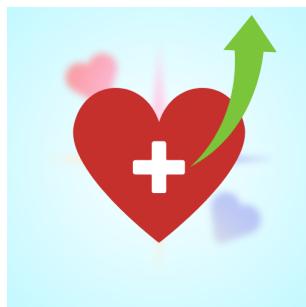
เป็นการโจมตีออกไปในแนวตรง สร้างความเสียหายให้กับศัตรูทุกตัวที่ผ่านทาง



รูปที่ 3.10: รูป Active Skill: Wind Blade

Increase Max Health

เป็น Passive Ability ที่จะช่วยเพิ่มพลังชีวิตสูงสุดของตัวละครผู้เล่น



รูปที่ 3.11: รูป Passive Skill: Increase Max Health

Increase Movement Speed

เป็น Passive Ability ที่จะช่วยเพิ่มความเร็วการเคลื่อนที่ของตัวละครผู้เล่น



รูปที่ 3.12: รูป Passive Skill: Increase Movement Speed

Increase Ability Damage

เป็น Passive Ability ที่จะช่วยเพิ่มความเสียหายของการโจมตีของตัวละครผู้เล่น



รูปที่ 3.13: รูป Passive Skill: Increase Ability Damage

Cooldown Reduction

เป็น Passive Ability ที่จะช่วยลดเวลา Delay ของ Active Ability



รูปที่ 3.14: รูป Passive Skill: Cooldown Reduction

3.1.6. ไอเทม

Exp Orb

เป็นค่าประสบการณ์ที่จะได้จากการกำจัดศัตรู เมื่อผู้เล่นไปสัมผัสแล้วเก็บสะสมครบที่กำหนดก็จะสามารถพัฒนาความสามารถของตัวละครผู้เล่นได้

Healing Orb

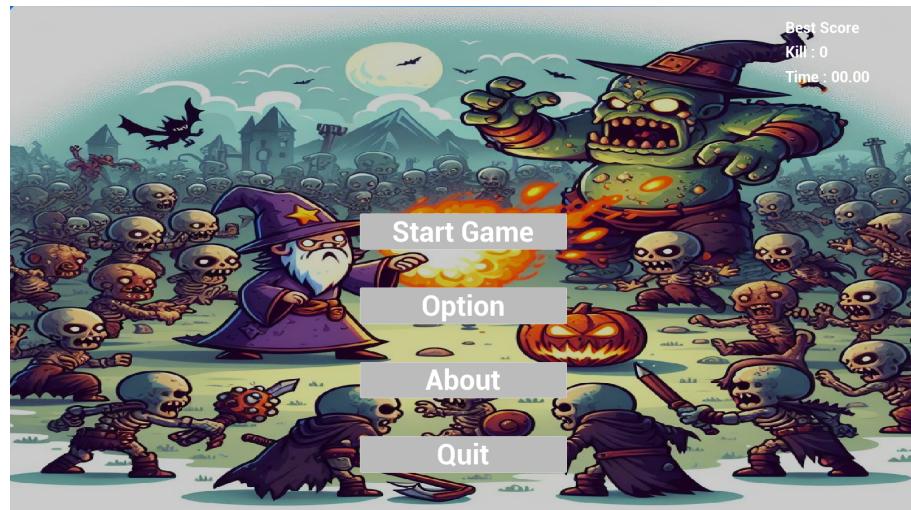
เป็นลูกแก้วที่เมื่อผู้เล่นไปสัมผัสแล้วจะฟื้นฟูพลังชีวิตที่ขาดหายไปได้จำนวนหนึ่ง

3.2. การออกแบบส่วนต่อประสานกับผู้ใช้

ในส่วนนี้คือการอธิบายถึงการออกแบบส่วนต่อประสานกับผู้ใช้โดยทางผู้จัดทำได้ออกแบบ หน้าส่วนต่อประสานใน Unreal Engine เพื่อให้เห็นภาพ โดยมีรายละเอียดดังนี้

3.2.1. หน้าเมนูหลัก

ในหน้านี้ผู้เล่นจะสามารถเลือกตัวเลือก Start Game, Option, About และ Quit ได้



รูปที่ 3.15: รูปหน้า Main Menu

3.2.2. หน้าเล่นเกม

หน้าจอเล่นเกมจะประกอบด้วย เวลาที่ผู้เล่นเอาชีวิตรอด, หลอดเลือดของผู้เล่น, หลอดประสบการณ์ของผู้เล่น



รูปที่ 3.16: รูปหน้าจอเล่นเกม

3.2.3. หน้าสรุปผลการเล่น

หน้าจอสรุปผลจะปรากฏขึ้นหลังจากผู้เล่นตายหรืออยู่รอดจนครบเวลาที่กำหนด ประกอบด้วยข้อมูลการเล่นต่างๆ เช่น เวลาที่ผู้เล่นเอาชีวิตรอดได้, จำนวนการกำจัดศัตรู, ปุ่ม Main Menu, ปุ่ม Restart



รูปที่ 3.17: รูปหน้าสรุปผลการเล่น

บทที่ 4

การพัฒนาระบบ

เนื้อหาในบทที่ 4 เป็นส่วนที่อธิบายแนวทางในการพัฒนาระบบโดยใช้ Unreal Engine ซึ่งเป็นโปรแกรมที่อำนวยความสะดวกในการสร้างเกมโดยใช้ Blueprint ในการเขียน Script เพื่อควบคุมการทำงานของเกม

4.1. การพัฒนาแผนที่ของเกม

ในการพัฒนาแผนที่ของเกมเริ่มต้นจากการสร้าง Landscape จากนั้นก็จัดการเรื่องแสงและเงา, ทำการ Paint Landscape, วาง Asset ที่เป็น Static mesh ต่างๆ ลงบนแผนที่เพื่อเพิ่มความหลากหลายของแผนที่



รูปที่ 4.1: แผนที่ของเกม

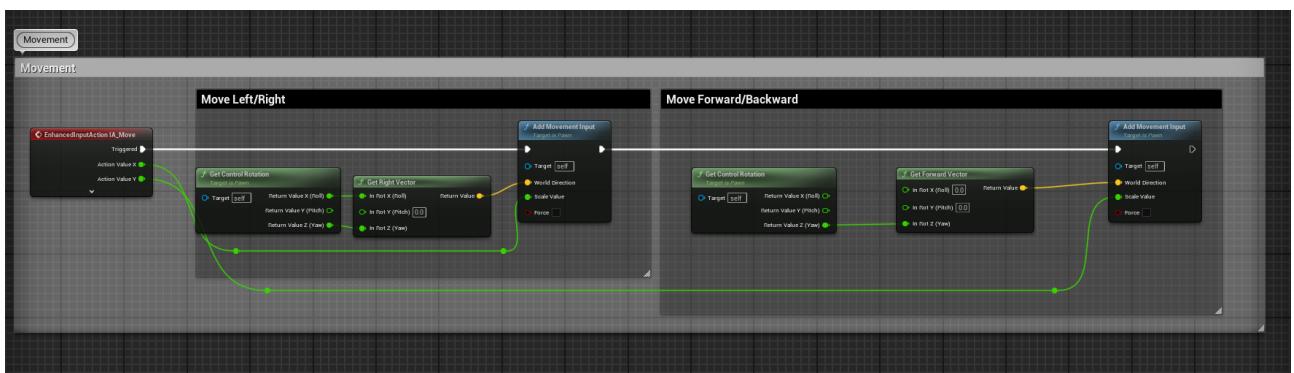
4.2. การพัฒนาตัวละครผู้เล่น

เริ่มต้นด้วยการสร้างโมเดลตัวละครและทำการใส่ Capsule Collision เพื่อตรวจจับการสัมผัสรึการชนของตัวละคร

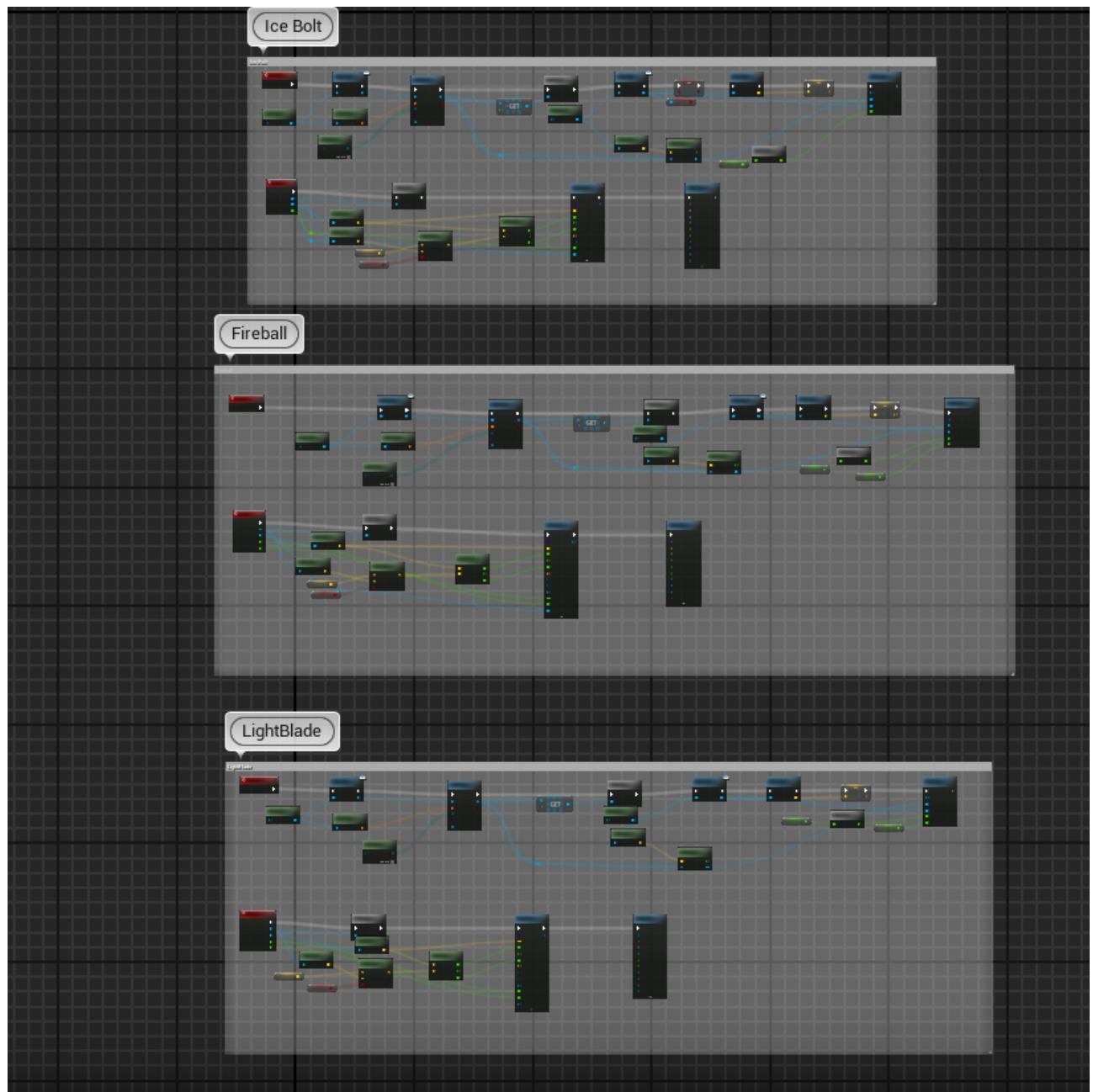


รูปที่ 4.2: หน้าสร้างตัวละคร

จากนั้นสร้าง Blueprint เพื่อควบคุมตัวละครไม่ว่าจะเป็นการโจมตี, การบังคับการเคลื่อนที่ของตัวละคร ด้วยการกดปุ่ม WASD, ระบบพลังชีวิตของตัวละคร, ระบบความสามารถของตัวละคร



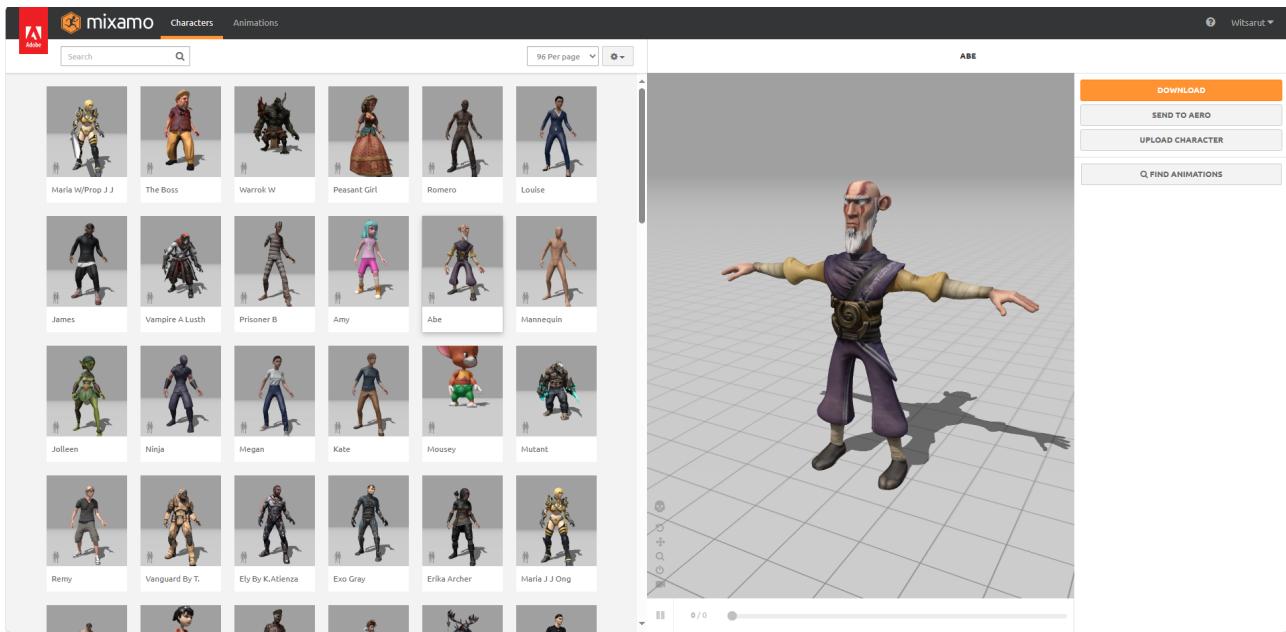
รูปที่ 4.3: Blueprint การบังคับตัวละคร



รูปที่ 4.4: Blueprint ความสามารถของตัวละคร

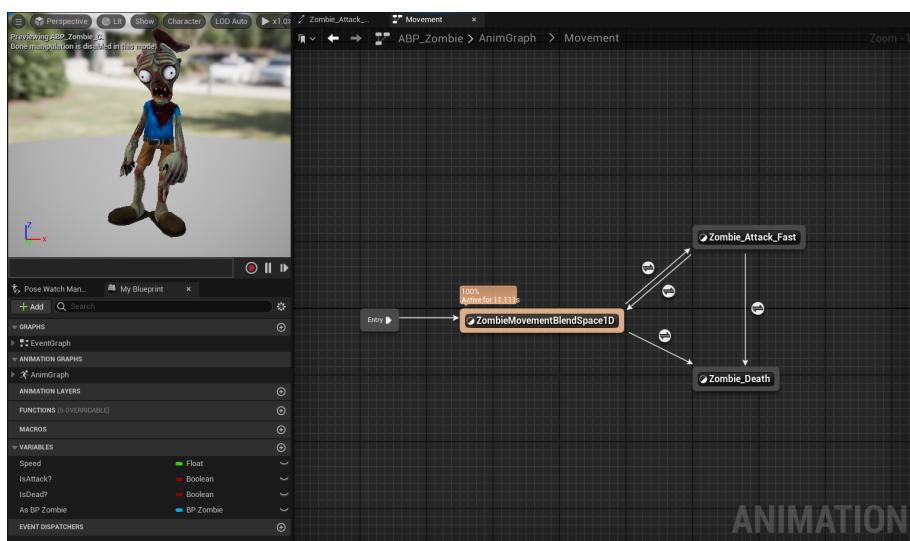
4.3. การพัฒนาศัตรู

เริ่มต้นจากการหาโมเดลตัวละครและอนิเมชันจากเว็บไซต์ Mixamo เพื่อที่จะนำมาออกแบบเป็นตัวละครศัตรู



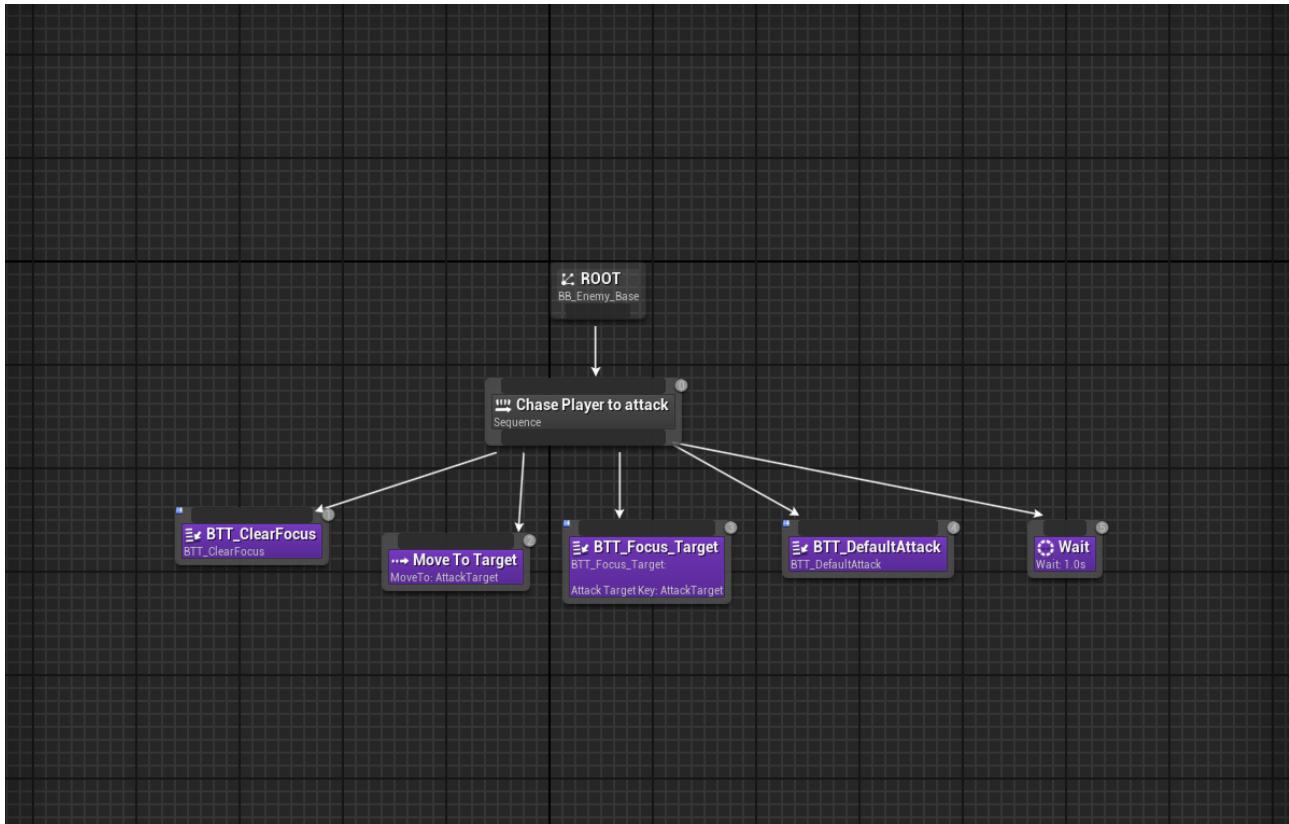
รูปที่ 4.5: หน้าเว็บไซต์ Mixamo

หลังจากนั้นก็จะนำโมเดลตัวละครและอนิเมชันที่ได้จากเว็บไซต์ Mixamo ทั้งหมดมาทำเป็น Animation Blueprint เพื่อควบคุมการเคลื่อนไหวของตัวละครศัตรู



รูปที่ 4.6: Animation Blueprint

หลังจากนั้นจะใช้ระบบ Behavior Tree ในกระบวนการคุ้มพัติกรรมและการตัดสินใจของศัตรู



รูปที่ 4.7: Behavior Tree

4.4. ระบบเกมเพย์

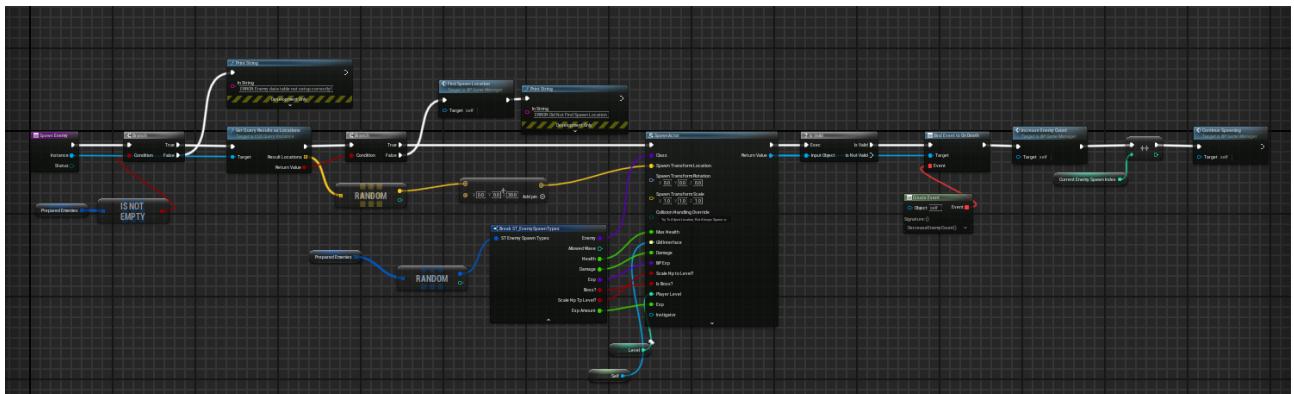
ระบบเกมเพย์จะเป็นระบบหลักของโปรเจค เป็นระบบที่ควบคุมการเล่นของผู้เล่น, การเกิดของศัตรู, การตั้งค่าต่างๆ ของเกม

4.4.1. ระบบการเกิดของศัตรู

ระบบจะทำการสุ่มเกิดศัตรูบริเวณรอบตัวของผู้เล่นอยู่ตลอดเวลา โดยข้อมูลต่างๆ ของศัตรูจะสามารถกำหนดได้ใน Data Table ที่ผู้จัดทำได้สร้างเอาไว้

Row Name	Enemy	Allowed Wave	Health	Damage	Exp	Boss?	Scale HpTpLevel	ExpAmount
1	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_0	40.000000	5.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	25.000000	
2	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_1	50.000000	7.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	35.000000	
3	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_2	60.000000	8.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	35.000000	
4	mageWave2 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_M_2	50.000000	10.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	40.000000	
5	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_3	70.000000	10.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	45.000000	
6	runnerWave3 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Ri_3	30.000000	7.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	45.000000	
7	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_4	50.000000	10.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	50.000000	
8	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_5	50.000000	10.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	50.000000	
9	mageWave5 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_M_5	70.000000	10.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	60.000000	
10	runnerWave6 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Ri_6	40.000000	10.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	80.000000	
11	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_7	80.000000	12.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	90.000000	
12	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_8	100.000000	15.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	90.000000	
13	runnerWave8 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Ri_3	45.000000	10.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	100.000000	
14	zombieWave1 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Zc_9	110.000000	15.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	150.000000	
15	runnerWave9 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_Ri_9	60.000000	15.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	180.000000	
16	mageWave9 /Script/Engine BlueprintGeneratedClass'/Game/Blueprints/Enemy/BP_M_9	80.000000	15.000000	/Script/Engine BlueprintGeneratedClass'/Game/Blueprints/BP_Exp_BP_E	False	False	200.000000	

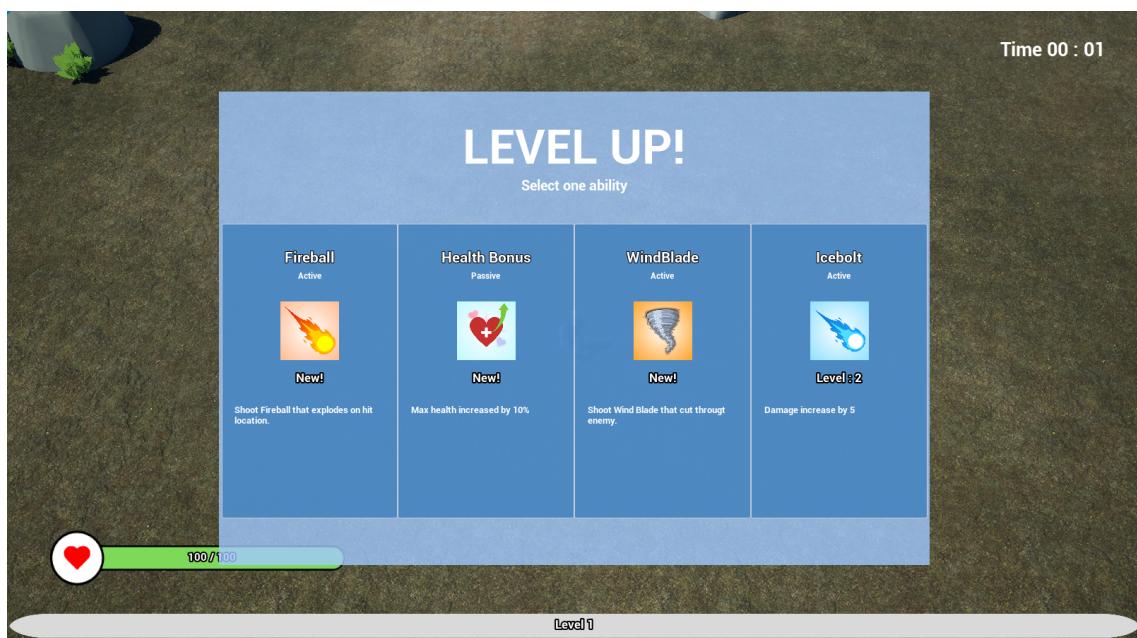
รูปที่ 4.8: Data Table ข้อมูลศัตรู



รูปที่ 4.9: Blueprint การเกิดของศัตรู

4.4.2. ระบบการอัพเกรดตัวละคร

ผู้เล่นสามารถเก็บค่าประสบการณ์จากการกำจัดศัตรู เพื่อนำมาเพิ่มความสามารถให้กับตัวละคร



รูปที่ 4.10: หน้าจอ Level Up

บทที่ 5

การทดสอบ ผลลัพธ์ และการประเมิน

การทดสอบเป็นสิ่งสำคัญที่ช่วยให้เราสามารถตรวจสอบการทำงานของส่วนต่างๆ ในระบบว่าสามารถทำงานได้ตรงตามวัตถุประสงค์ และไม่มีข้อผิดพลาด โดยจะทดสอบระบบตามส่วนดังนี้

5.1. ระบบตัวละครผู้เล่น

ในส่วนที่แสดงในรูปตารางที่ 5.1 จะเป็นการทดสอบการทำงานของระบบตัวละครผู้เล่นว่ามีข้อผิดพลาดหรือไม่

กรณีทดสอบ	เงื่อนไขก่อนทดสอบ	ผลการทดสอบ	หมายเหตุ
กดปุ่มควบคุมตัวละครถูกต้อง	ผู้เล่นกดปุ่ม WASD เพื่อทำการเคลื่อนที่	ตัวละครเคลื่อนไหวตามที่กำหนดไว้	-
ค่าพลังชีวิตของผู้เล่น	ผู้เล่นถูกโจมตีโดยศัตรู	พลังชีวิตของผู้เล่นลดลงอย่างถูกต้อง	-
ค่าประสบการณ์ของผู้เล่น	ผู้เล่นเก็บค่าประสบการณ์จากการกำจัดศัตรู	หลอดแสดงค่าประสบการณ์ของผู้เล่นเพิ่มขึ้นได้ถูกต้อง	-

รูปที่ 5.1: ตารางแสดงการทดสอบการทำงานของตัวละครผู้เล่น

5.2. ระบบศัตรู

ในส่วนที่แสดงในรูปตารางที่ 5.2 จะเป็นการทดสอบการทำงานของระบบศัตรูว่ามีข้อผิดพลาดหรือไม่

กรณีทดสอบ	เงื่อนไขก่อนทดสอบ	ผลการทดสอบ	หมายเหตุ
การเกิดของศัตรู	ผู้เล่นเข้าสู่เกม	การสุ่มเกิดของศัตรูถูกต้อง	-
พฤติกรรมของศัตรูถูกต้อง	ศัตรูเข้ามาโจมตีผู้เล่น	พฤติกรรมของศัตรูถูกต้อง	-

รูปที่ 5.2: ตารางแสดงการทดสอบการทำงานของศัตรู

5.3. ระบบสิ่งแวดล้อม

ในส่วนที่แสดงในรูปตารางที่ 5.3 จะเป็นการทดสอบการทำงานของระบบสิ่งแวดล้อมว่ามีข้อ ผิดพลาดหรือไม่

กรณีทดสอบ	เงื่อนไขก่อนทดสอบ	ผลการทดสอบ	หมายเหตุ
ผู้เล่นเดินชนจาก	ตัวละครของผู้เล่นเดินชนสิ่งกีดขวาง	ตัวละครของผู้เล่นไม่หลบสิ่งกีดขวาง	-
ผู้เล่นโจมตีโคนจาก	Projectile ที่ผู้เล่นสร้างชนกับสิ่งกีดขวาง	Projectile ของผู้เล่นไม่หลบจาก	-
ผู้เล่นได้ยินเสียงเกม	ผู้เล่นได้ยินเสียงเอฟเฟคต่างๆ	ผู้เล่นได้ยินเสียงอย่างถูกต้อง	

รูปที่ 5.3: ตารางแสดงการทดสอบการทำงานของสิ่งแวดล้อม

บทที่ 6

บทสรุป ปัญหา และข้อเสนอแนะ

6.1. บทสรุป

การพัฒนาเกมต่อสู้อาชีวิตรอด โดยใช้ Unreal Engine นั้นมีวัตถุประสงค์ในการสร้าง ความท้าทาย โดยการสุ่มตำแหน่งการเกิดของศัตรู และสุ่มเลือกการพัฒนาตัวละคร อีกทั้งยัง ใช้ระบบ Behavior Tree และ AI Perception ในการควบคุมการทำงานของศัตรู

โดยในการพัฒนาเกมได้มีการใช้ Unreal Engine เป็นเครื่องมือหลักในการพัฒนา โปรเจคด้วยการใช้ระบบ Blueprint และได้มีการใช้เว็บไซต์ Mixamo เป็นเครื่องมือในการ ออกแบบอนิเมชันของตัวละครต่างๆ ที่ใช้ภายในเกม

6.2. ปัญหาและข้อเสนอแนะ

ในการพัฒนาเกมในครั้งนี้พบปัญหาเรื่องการวางแผนในการทำงาน เนื่องจากเป็นการพัฒนา เกมด้วย Unreal Engine ครั้งแรกของผู้จัดทำ จึงทำให้ใช้เวลาส่วนใหญ่ไปกับการศึกษาการใช้งานโปรแกรม และ การแก้ไขปัญหาที่เกิดขึ้นจากความไม่เข้าใจการทำงานของโปรแกรม ซึ่งทำให้เกิดความล่าช้าในการทำงาน

ในส่วนของตัวเกมยังพบปัญหาเกี่ยวกับการแสดงผลอนิเมชันที่ต่อเนื่องกัน ทำให้อาจจะดูไม่ต่อเนื่องกันในบางเหตุการณ์

6.3. แนวทางในการพัฒนาต่อไป

เกมแนวนี้ควรที่จะมีความหลากหลายในการเล่นมากขึ้น ควรจะมีการเพิ่มรูปแบบการเล่นมากขึ้น เช่น เพิ่มรูปแบบของศัตรู เพิ่มความสามารถของตัวละคร ดังนั้นผู้พัฒนาจึงจะศึกษาค้นคว้าเพิ่มเติมเกี่ยวกับการใช้งานโปรแกรม และเพิ่มทักษะการสร้างเกม ให้ดีขึ้น อีกในอนาคต

បរណាន់ករណ

- Aulén, J. (2020). Animation state machines in unreal engine.
- Båtelsson, H. (2016). Behavior trees in the unreal engine: Function and application.
- Bethke, E. (2003). *Game development and production*. Wordware Publishing, Inc.
- Blackman, S. and Blackman, S. (2014). Rigging with mixamo. *Unity for Absolute Beginners*, pages 565–573.
- Bourg, D. M. and Seemann, G. (2004). *AI for game developers*. ” O'Reilly Media, Inc.”.
- Josuttis, N. M. (2012). The c++ standard library: a tutorial and reference.
- Kushnir, V. and Koman, B. (2018). Creating ai for games with unreal engine 4. *Electronic Journal of Information Technology*, 9.
- Lee, J. (2016). *Learning unreal engine game development*. Packt Publishing Ltd.
- Sanders, A. (2016). *An introduction to Unreal engine 4*. AK Peters/CRC Press.
- Valcasara, N. (2015). *Unreal engine game development blueprints*. Packt Publishing Ltd.