โครงงานเลขที่ วศ.คพ. S049-1/2566

เรื่อง

ตรวจสอบประกาศนียบัตรออนไลน์ด้วยบล็อคเชน

โดย

นายคนธกานต์ ฟูคำ รหัส 630610719
นายคุณาสิน เตชะสืบ รหัส 630610721

โครงงานนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2566

PROJECT No. CPE S049-1/2566

E-Certificate using Hyperledger Fabric Blockchain

Konthakarn Fukam 630610719
Kunasin Techasueb 630610721

A Project Submitted in Partial Fulfillment of Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering
Faculty of Engineering
Chiang Mai University
2023

หัวข้อโครงงาน	: ตรวจสอบประกาศนียบัตรออนไลน์ด้วยบล็อคเชน
	: E-Certificate using Hyperledger Fabric Blockchain
โดย	: นายคนธกานต์ ฟูคำ รหัส 630610719
	นายคุณาสิน เตชะสืบ รหัส 630610721
ภาควิชา	: วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	: รศ.ดร. ตรัสพงศ์ ไทยอุปถัมภ์
ปริญญา	: วิศวกรรมศาสตรบัณฑิต
สาขา	: วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	: 2566

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กิจนุกูร)

คณะกรรมการสอบโครงงาน

..... ประธานกรรมการ
(รศ.ดร. ตรัสพงศ์ ไทยอุปถัมภ์)

..... กรรมการ
(ผศ. โดม โพธิกานนท์)

..... กรรมการ
(ผศ.ดร. กำพล วรดิษฐ์)

หัวข้อโครงงาน	: ตรวจสอบประกาศนียบัตรออนไลน์ด้วยบล็อคเชน
	: E-Certificate using Hyperledger Fabric Blockchain
โดย	: นายคนธกานต์ ฟูคำ รหัส 630610719
	นายคุณาสิน เตชะสืบ รหัส 630610721
ภาควิชา	: วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	: รศ.ดร. ตรัสพงศ์ ไทยอุปถัมภ์
ปริญญา	: วิศวกรรมศาสตรบัณฑิต
สาขา	: วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	: 2566

บทคัดย่อ

ในปัจจุบัน ประกาศนียบัตร (Certificate) จากการศึกษา หรือฝึกอบรม มีแนวโน้มที่จะเป็นรูปแบบกระดาษ (Paper-based) ลดน้อยลง โดยมีการเปลี่ยนไปใช้รูปแบบอิเล็กทรอนิกส์มากขึ้นเรื่อย ๆ ปัญหาของการปลอมแปลงประกาศนียบัตรยังคงมีอยู่ และมีแนวโน้มที่จะรุนแรงมากขึ้นถึงแม้ว่าจะเป็นรูปแบบอิเล็กทรอนิกส์ก็ตาม เราไม่สามารถรู้ได้เลยว่าประกาศนียบัตรนั้นเป็นของจริงหรือของปลอม จากการสำรวจพบว่ามีซอฟต์แวร์ในการปลอมแปลงเอกสารเหล่านี้เกิดขึ้นมาจำนวนมากในโลกออนไลน์ และพบว่าร้อยล่ะ30ของทั่วโลกมีการปลอมคุณสมบัติขึ้นที่ตนไม่มีขึ้นมา เราจึงต้องระวังข้อมูลปลอมและหาวิธีการยืนยันเอกสารเหล่านั้นให้ได้ ในโครงงานนี้จึงมีแนวคิดในการออกแบบและพัฒนาระบบประกาศนียบัตรอิเล็กทรอนิกส์โดยใช้ไฮเปอร์เลจเจอร์เฟบริกบล็อคเชน เพื่อเพิ่มความน่าเชื่อถือ ความถูกต้อง และความปลอดภัยในการรับรอง การยืนยัน และการตรวจสอบประกาศนียบัตร (Certificate) ที่ออกโดยสถานศึกษา ซึ่งเทคโนโลยีบล็อกเชนมีคุณลักษณะพื้นฐานที่สามารถเก็บข้อมูลได้อย่างปลอดภัย มีการใช้ Cryptography ในการป้องกันการแก้ไขข้อมูลในข้อมูลที่เก็บเป็นลักษณะของบล็อกที่เชื่อมต่อกันเป็นเชน โดยโครงงานนี้ใช้เทคโนโลยีไฮเปอร์เลจเจอร์เฟบริกบล็อคเชน ที่เป็นบล็อกเชนที่ต้องได้รับอนุญาตในการเข้าร่วม (Permissioned Blockchain) มีการใช้ระบบ Digital Signature ในการยืนยันตัวตนในการเข้าถึงข้อมูลผ่าน Smart Contract (Chaincode) เพื่อความปลอดภัย และความน่าเชื่อถือของข้อมูลที่เก็บในบล็อกเชน

Project Title	: E-Certificate using Hyperledger Fabric Blockchain
Name	: Konthakarn Fukam 630610719
	Kunasin Techasueb 630610721
Department	: Computer Engineering
Project Advisor	: Assoc. Prof. Trasapong Thaiupathump, Ph.D.
Degree	: Bachelor of Engineering
Program	: Computer Engineering
Academic Year	: 2023

ABSTRACT

Currently, there is a trend towards reducing the use of paper-based certificates for education or training, with an increasing shift towards electronic formats. However, the problem of certificate forgery persists, and it is becoming more severe even in electronic formats. We cannot always be certain whether a certificate is genuine or fake.

It has been found that there is a significant amount of software for forging these documents available online, and approximately 30 percent of the global population has experienced the falsification of their credentials. Therefore, we must be cautious about counterfeit information and find ways to verify these documents.

In this project, the idea is to design and develop an electronic certificate system using Hyperledger Fabric blockchain technology to enhance trustworthiness, accuracy, and security in certifying, verifying, and validating certificates issued by educational institutions. Blockchain technology has fundamental features that allow secure data storage, utilizing cryptography to prevent data tampering within interconnected blocks. This project uses a permissioned blockchain, which requires authorization to participate. It also incorporates digital signature systems for identity verification when accessing data through Smart Contracts (Chaincode) to ensure data security and trustworthiness within the blockchain.

กิตติกรรมประกาศ

โครงงานนี้จะสำเร็จลุล่วงได้ด้วยความกรุณาจากอาจารย์ รศ.ดร.ตรัสพงศ์ ไทยอุปถัมภ์ อาจารย์ที่ปรึกษาที่ได้เสียสละเวลาอันมีค่าแก่นักศึกษาโครงงานนี้ได้รับข้อเสนอแนะและแนวคิดตลอดจนการแก้ไขข้อบกพร่อง รายละเอียดต่างๆตรวจทานแก้ไขด้วยความเอาใจใส่เป็นอย่างยิ่งจนโครงการฉบับนี้สำเร็จสมบูรณ์ลุล่สงไป ได้ด้วยดีขอกราบของพระคุณเป็นอย่างสูงไว้ณที่นี้จากใจจริงรวมถึง ผศ.โดม โพธิกานนท์ และ ผศ.ดร.กำพล วรดิษฐ์ ที่ให้คำปรึกษา คำแนะนำ จนทำให้โครงงานเล่มนี้มีความสมบูรณ์มากที่สุด

ขอบคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่ให้สถานที่ในการทำโครงงาน ทั้งห้องภาควิชาวิศวกรรมคอมพิวเตอร์ และสถานที่ต่างๆในภาควิชา และยังให้การสนับสนุนทางด้านงบประมาณ อุปกรณ์ต่างๆ ที่จำเป็นต่อการทำโครงงาน

ขอขอบพระคุณผู้ปกครอง เพื่อนๆ และรุ่นพี่ทุกๆคน ที่ให้คำปรึกษา คำแนะนำ และคอยเป็นกำลังใจให้ตลอดมา ซึ่งเป็นแรงผลักดันให้แก่ผู้จัดทำมีความตั้งใจและมุ่งมั่นในการทำงาน จนโครงงานที่ความสมบูรณ์มากที่สุด

นอกจากนี้ผู้จัดทำขอขอบพระคุณอีกหลายๆท่านที่ไม่ได้กล่าวถึง ณ ที่นี้ ที่ได้ให้ความช่วยเหลือตลอดมา และสุดท้ายนี้ หากโครงงานนี้มีข้อผิดพลาดประการใด ผู้จัดทำขออภัยมา ณ ที่นี้ และพร้อมน้อมรับด้วยความยินดี

นายคนธกานต์ ฟูคำ
นายคุณาสิน เตชะสืบ
5 ตุลาคม 2566

สารบัญ

สารบัญรูป

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

เกมเป็นสื่อบันเทิงประเภทหนึ่งที่มีการแพร่หลายเป็นอย่างมากในปัจจุบัน ไม่ว่าจะเป็นเกมบนมือถือ บนเว็บไซต์ เกมบนเครื่องเล่นเกมต่างๆที่ออกแบบมาเพื่อเกมใดเกมหนึ่งโดยเฉพาะ และรวมไปถึงเกมบนคอมพิวเตอร์ ซึ่งบางเกมได้มีการจัดการแข่งขันกันขึ้น เพื่อชิงรางวัลต่างๆมากมายภายในงานแข่ง ส่งผลให้ผู้คนให้ความสนใจกับเกมมากขึ้น และส่งผลให้อุสาหกรรมเกมมีการเติบโตอย่างรวดเร็ว จนเกิดอาชีพใหม่ๆมากมายที่เกี่ยวกับเกม เช่น Streamer, นักกีฬา E-sport, นักพากย์เกม เป็นต้น

โดยโครงงานนี้ก็ได้เริ่มต้นมาจากการที่ผู้พัฒนาชื่นชอบในการเล่นเกม และมีความสนใจที่จะสร้างเกมขึ้นมาหนึ่งเกม ซึ่งผู้พัฒนาได้ลองทำการศึกษาพื้นฐานต่างๆเกี่ยวกับการสร้างเกม และตัดสินใจเสนอความสนใจเหล่านี้พร้อมกับอธิบายเหตุผลให้กับอาจารย์ฟัง จนสุดท้ายได้ทำการตกลงกับอาจารย์ว่าจะสร้างเกม 3D แนว RPG action ขึ้นมา ซึ่งก็คือ เกม Miracle from sky นั่นเอง

สำหรับเกม Miracle from sky เป็นเกมแนว Action RPG OpenWorld แบบ Single-player ที่มีมุมมองเป็น มุมมองของบุคคลที่สาม ซึ่งผู้พัฒนาให้ความสนใจ และอยากนำมาเป็นต้นแบบในการทำเกมคือ Genshin impact และ Diablo ซึ่งทางระบบ gameplay จะเน้นไปทาง Genshin impact ส่วนระบบสกิลจะเน้นไปทาง Diablo ซึ่งในเกม ผู้เล่นจะได้รับบทเป็นเด็กสาวที่ต้องผจญภัยในโลกกว้าง และฝึกฝนตัวเองให้เก่งขึ้น เพื่อที่จะไปปราบจอมมาร โดยระหว่างการเดินทางผู้เล่นจะได้พบศัตรูหลากหลายรูปแบบ ซึ่งต้องใช้วิธีรับมือที่แตกต่างกัน ได้สำรวจโลกแฟนตาซีกว้างใหญ่ และได้พบเจอกับปริศนาต่างๆที่รอให้ผู้เล่นได้เข้าไปแก้ไขหาคำตอบ

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อตอบสนองความสนใจ ความต้องการของผู้พัฒนาที่อยากจะทำเกมของตัวเองขึ้นมาซักหนึ่งเกม
2. เพื่อสร้างประสบการณ์ต่างๆที่น่าตื่นเต้น สนุกสนาน และน่าติดตามให้กับผู้เล่น ผ่านทางตัวเกม ทั้งด้านเนื้อเรื่อง gameplay และสิ่งต่างๆภายในเกม
3. เพื่อสร้างความบันเทิงให้กับผู้เล่น และช่วยทำให้ผู้เล่นรู้สึกผ่อนคลายเวลาเล่นเกม
4. เพื่อเป็นแบบอย่าง และแรงบรรดาลใจให้กับหลายๆคนที่อยากจะลองสร้างเกมของตัวเองขึ้นมา

1.3 ขอบเขตของโครงงาน

1.3.1 ขอบเขตด้านฮาร์ดแวร์

1. เกมสามารถเล่นได้ผ่านทางคอมพิวเตอร์ ซึ่งได้แก่ PC, laptop
2. เกมจะใช้เมาส์ แป้นพิมพ์ ในการควบคุม

1.3.2 ขอบเขตด้านซอฟต์แวร์

1. เกมจะรองรับแค่ระบบปฏิบัติการ Windows

2. เกมถูกออกแบบมาสำหรับผู้เล่นคนเดียว
3. เกมมีมุมมองเป็นแบบมุมมองบุคคลที่สาม เท่านั้น ไม่สามารถเปลี่ยนมุมมองอื่นๆได้

1.4 ประโยชน์ที่ได้รับ

1. ผู้เล่นจะได้รับความสนุกสนาน ความบันเทิงต่างๆภายในเกม
2. ผู้เล่นจะได้รับประสบการณ์ใหม่ๆมากมายจากการเกม
3. ผู้พัฒนาได้รับประสบการณ์ใหม่ๆในการทำงานเป็นทีม และประสบการณ์ต่างๆในการสร้างเกม ซึ่งเป็นผลดีต่อการทำงานในอนาคต

1.5 เทคโนโลยีและเครื่องมือที่ใช้

1.5.1 เทคโนโลยีด้านฮาร์ดแวร์

1. คอมพิวเตอร์รุ่น Asus zephyrus g14, Ryzen 7 ใช้ในการออกแบบ และพัฒนาเกม
2. คอมพิวเตอร์รุ่น Acer aspire5, core i7 ใช้ในการออกแบบ และพัฒนาเกม โดยจะใช้คอมเครื่องนี้เป็นตัวหลักในการสร้างโปรเจคหลัก

1.5.2 เทคโนโลยีด้านซอฟต์แวร์

1. ระบบปฏิบัติการ Windows โดยจะใช้เป็น Window 10
2. Unity ใช้เป็นตัวหลักในการพัฒนาเกม โดยจะใช้ platform 3D ของ Unity ในการสร้างเกม
3. Visual studio ใช้ในการเขียน script ควบคุมระบบเกมต่างๆ ซึ่งใช้ภาษา C# ในการเขียน code
4. Blender ใช้ในการสร้างโมเดล 3D สำหรับใช้ในเกม
5. Krita ใช้ในการออกแบบและ ช่วยในการสร้างเอฟเฟคต่างๆในเกม
6. Photoshop ใช้ในการออกแบบและ ช่วยในการสร้างเอฟเฟคต่างๆในเกม

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ส.ค. 2566	ก.ย. 2566	ต.ค. 2566	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567	มี.ค. 2567
ศึกษาค้นคว้าการใช้งาน Unity								
ศึกษาค้นคว้าการใช้ Visual studio ในการเขียน script								
วางแผนออกแบบเกม เช่น ออกแบบระบบต่างๆ เนื้อเรื่อง แผนที่ ปริศนาต่าง ตัวละครที่จะใช้ เป็นต้น								
ระบบควบคุม เช่น การเดิน การกระโดด มุมกล้อง เป็นต้น								

ขั้นตอนการดำเนินงาน	ส.ค. 2566	ก.ย. 2566	ต.ค. 2566	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567	มี.ค. 2567
ระบบต่อสู้ เสียง การแสดงท่าทางเมื่อโจมตี								
สร้างmap รวมถึงจัดสถานที่บอส ปริศนาต่างๆ และว่างเนื้อเรื่อง								
effect และ UI ต่างๆในเกม เช่น หลอดเลือด level เป็นต้น								
รวมทุกอย่างเข้าด้วยกันให้เกมสามารถเล่นจนจบเกมได้								
รวบรวมข้อมูล ความสนใจของผู้เล่น และทำเล่มโครงงาน								
ตรวจสอบความถูกต้องของโครงงาน รวมถึงบัคต่างๆในเกม และตกแต่งเกมให้มีความสมบูรณืมากยิ่งขึ้น								

1.7 บทบาทและความรับผิดชอบ

สำหรับการแบ่งงานของกลุ่มของพวกเราจะแบ่งออกเป็น 3 ส่วนหลักๆ

1.การออกแบบเกมโดยรวม: ในส่วนนี้จะช่วยกันทำ โดยการระดมความคิด ข้อเสนอต่างๆ มารวมกันแล้วเลือกเอาในสิ่งที่ สามารถทำได้และ สิ่งเห็นตรงกันว่าอยากจะให้มีในเกมของพวกเรา

2.การออกแบบต่างๆ: สำหรับการออกแบบส่วนใหญ่รับผิดชอบโดย นายเทวฤทธิ์ สมฤทธิ์ ไม่ว่าจะเป็นแมพ ปริศนาต่างๆ effect เสียงต่าง แต่โดยรวมแล้วช่วยๆกันทำ โดยเฉพาะการเลือกตัวละคร และเนื้อเรื่องของเกม

3.การออกแบบระบบเกม: สำหรับการออกแบบระบบเกมรับผิดชอบโดย นายชาญชัย ไชยสลี ไม่ว่าจะเป็นระบบควบคุมตัวละคร การออกท่าโจมตี ระบบเลือด แต่โดยรวมแล้วช่วยๆกันทำ โดยเฉพาะระบบกาควบคุมตัวละคร ระบบการอัพเลเวล การต่อสู้

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

สำหรับเกม Miracle from sky เป็นเกมที่เน้นสร้างความสนุกสนาน ความบันเทิงให้กับผู้เล่น ดังนั้นภายในเกมไม่มีฉากที่ล่อแหลม ทั้งทางเพศ และทางการกระทำผิดกฎหมาย และเนื่องจากเกมของพวกเราเน้นให้ผู้เล่นผ่อนคลาย ไม่ว่าจะเด็กหรือผู้ใหญ่ ดังนั้นในเกมจึงไม่มีการใส่ effect เลือดสาดต่างๆเข้าไป แต่ถึงอย่างนั้น ในเกมก็ยังมาฉากการตายของตัวละครหลัก และมอนสเตอร์ รวมถึงมีการต่อสู้ มีใช้อาวุธต่างๆ เพื่อใช้ในการฆ่าศัตรู

เกม Miracle from sky สามารถนำไปเป็นต้นแบบ แนวทาง หรือแรงบันดาลใจ ในการทำเกมแนว action RPG OpenWorld ได้ นอกจากนั้น โครงงานนี้ยังสามารถนำไปประยุกต์ใช้งานร่วมกับโครงงานอื่นได้ เช่น โครงงานที่ศึกษาเกี่ยวกับการลดความเครียดโดยการเล่นเกม โครงงานที่ศึกษาเกี่ยวกับการเล่นเกมว่าจะส่งผลกระทบอะไรกับเรียนรู้ของเด็กบ้าง เป็นต้น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

Miracle from sky ถูกสร้าง และพัฒนาขึ้นโดยโปรแกรม Unity เป็นหลัก ซึ่งก่อนที่ผู้พัฒนาจะเริ่มลงมือสร้างเกมจริงขึ้นมา ผู้พัฒนาได้ศึกษาหาความารู้ในด้านต่างๆที่จำเป็นสำหรับการสร้างเกม โดยเนื้อหาในบทนี้จะอธิบายในส่วนของความรู้ ทฤษฎีบทที่เกี่ยวข้อง และหลักการต่างๆที่ผู้พัฒนาได้ศึกษา และนำไปใช้ในการสร้างเกม เพื่อให้ผู้ที่เข้ามาอ่านได้เข้าใจหลักการต่างๆในเบื้องต้น และเพื่อให้เข้าใจเนื้อหาในบทถัดๆไปได้ง่ายมากยิ่งขึ้น

2.1 พื้นฐาน Unity

Unity เป็น software ที่ถูกออกแบบมาเพื่อใช้สำหรับการพัฒนา software ที่สามารถ จำลองการทำงานต่างๆได้ เช่น game(ทั้ง 2D และ 3D), การขนส่ง, animation, อุตสาหกรรมยานยนต์ เป็นต้น ซึ่งสามารถรองรับได้หลากหลาย platform เช่น PC, iOS, Android เป็นต้น โดยในที่นี้จะขออธิบายในส่วนที่เกี่ยวข้องกับการสร้าง game 3D เท่านั้น

โดยส่วนต่างๆใน Unity ที่สำคัญ และจำเป็นต้องศึกษาสำหรับการสร้างเกม มีดังนี้

2.1.1 Scene

คือ ฉากภายในเกม หรือบริเวณที่เรานำสิ่งต่างๆมาใช้รวมกัน ซึ่ง scene มีได้หลาย scene เช่น scene เริ่มเกม, scene จบเกม, scene เมนู เป็นต้น

2.1.2 GameObject

คือ วัตถุ หรือสิ่งต่างๆที่สามารถนำมาใช้แสดงผลภายใน scene ได้ เช่น Model ต่างๆ, ตัวเล่นเสียง, ตัวเล่น effect, light, terrain เป็นต้น

2.1.3 Asset

คือ GameObject หรือสิ่งต่างๆที่นำเข้ามาใช้งานใน project ของเรา เช่น Model ตัวละคร, เสียง, animation, script, texture, prefab, terrain เป็นต้น โดย asset เราสามารถซื้อจาก Unity Asset Store ได้ ซึ่งมีทั้ง ที่แจกฟรี และเสียเงิน โดยราคาขึ้นกับคุณภาพของ asset และความพึงพอใจของผู้ขาย

2.1.4 Camera

คือ กล้องที่ใช้สำหรับการแสดงผลเกมของเราออกมาให้ผู้เล่นเห็นทางจอภาพ โดยสามารถปรับมุมมอง ตำแหน่งต่างของกล้องได้อย่างอิสระ สามารถตั้งให้กล้องติดตามตัวผู้เล่นได้ รวมไปถึงใช้ในการทำ cutscene

2.1.5 Light

คือ GameObject ประเภทหนึ่งที่สามารถให้แสงสว่างกับ scene ของเราได้ light ทำให้เกิดเงาของ GameObject ซึ่งสามารถไปปรับใช้งานได้หลากหลาย เช่น ทำเวลากลางวัน/กลางคืน, ทำ scene มืดๆที่ทำให้รู้สึกถึงความน่ากลัว เป็นต้น

2.1.6 Component

คือ คุณสมบัติ หรือความสามารถต่างๆที่อยู่ใน GameObject ซึ่งมีหลากหลายคุณสมบัติ และคุณสมบัติแต่ละตัวก็มีความแตกต่างกันไป โดย component ที่สำคัญมีดังนี้

1. Transform คือ component ที่ใช้ในการควบคุมตำแหน่ง(Position) การหมุน(Rotation) และขนาด(Scale) โดยทุก GameObject ต้องมีคุณสมบัตินี้
2. Rigidbody คือ component ที่ใช้ในการจัดการเกี่ยวกับระบบฟิสิกส์ของวัตถุ ไม่ว่าจะเป็น แรง(Force), มวล(Mass), การแสดงผลจากแรงโน้มถ่วง(Gravity) และการล็อควัตถุ(Freeze)
3. Collider คือ component ที่ใช้ในการตรวจสอบการชนกันของวัตถุต่างๆภายใน scene นำมาประยุกต์ใช้ได้หลากหลายแบบ เช่น การคำนวณดาเมจ, การระเบิดของลูกบอลไฟเมื่อชนกับวัตถุต่างๆ, การเก็บไอเทมต่างๆ เป็นต้น
4. Animator คือ component ที่ใช้ในการควบคุมการทำงานของ animation ต่างๆ โดยจะควบคุม และแสดงในรูปของ state machine
5. Particle System คือ component ที่ใช้ในการสร้าง visial effect [5] หรือที่เรียกว่า VFX เช่น เปลวไฟ, สายฟ้า, น้ำ เป็นต้น
6. Volume คือ component ที่ใช้ในการควบคุมการแสดงผลทางหน้าจอ หรือภาพที่เราเห็นผ่านทางกล้อง โดยสามารถปรับบริมาณแสงที่ผ่านกล้อง, การเบลอขอบจอภาพ, การเพิ่มมุมแบบ perspective ทำให้ภาพที่เห็นถูกยืด หรือหดลงได้

2.1.7 Texture

คือ รูปภาพพื้นผิวต่างๆ ที่ใช้ในการนำมาเป็นผิวของวัตถุ ช่วยให้วัตถุมีความสมจริงมากขึ้น

2.1.8 Material

คือ เม็ดสี หรือสีที่ใช้ลงสีให้กับวัตถุต่างภายใน scene ไม่จำเป็นต้องเป็นสีล้วน โดยถ้าหากใช้ shader graph ในการสร้าง material จะทำให้ material ที่ได้มีคุณสมบัติที่กำหนดไว้ได้ เช่น material ที่เรืองแสงได้, material ที่มีความมันวาว, material ที่เปลี่ยนรูปร่างได้ เป็นต้น

2.1.9 SkyBox

คือ สิ่งที่ให้เปลี่ยน สี รูปแบบ คุณสมบัติต่างๆของท้องฟ้าภายในเกม เช่น สามารถทำให้ท้องฟ้าเป็นกลางคืน/วันได้, ทำให้ท้องฟ้ามีก้อนเมฆได้ เป็นต้น

2.1.10 Wind Zone

คือ GameObject ประเภทหนึ่ง ทำหน้าที่ช่วยควบคุมการทำงานของระบบลมภายใน scene ช่วยให้เกมมีความสมจริงมากยิ่งขึ้น

2.1.11 Terrain

คือ GameObject ประเภทหนึ่ง ทำหน้าที่ช่วยควบคุม ปรับแต่งภูมิประเทศ หรือสภาพแวดล้อมของพื้น ให้มีลักษณะตามที่เราต้องการ เช่น ใช้ทำหลุม, ใช้ทำภูเขา, ใช้ทำพื้นที่ยกระดับ เป็นต้น

2.1.12 Prefab

คือ การนำ GameObject ต่างๆมาประกอบกันเพื่อสร้างเป็น GameObject ใหม่ที่รวม GameObject หลายๆตัวเอาไว้ ซึ่งจะมีลักษณะ ต่างๆตามที่เรากำหนด

2.1.13 Tag

คือ สิ่งที่ใช้กำหนด หรือจำแนกประเภทของ GameObject ตามที่เรากำหนด จะใช้ประโยชน์ในการตรวจสอบว่า GameObject นี้คืออะไร เช่น สร้าง tag ชื่อ enemy กับ tag ชื่อ player เพื่อใช้ในการระบุว่า GameObject นี้คือ enemy หรือ player เป็นต้น

2.1.14 Layer

คือ ลำดับชั้นการแสดงผล รวมถึงการทำงานของวัตถุ โดย layer สูงๆ หรือ layer ที่มีเลขต่ำๆ จะมีสิทธิ์ถูกสั่งให้แสดงผล หรือ ได้ทำงานก่อนเป็นลำดับแรกๆ

2.1.15 Script

คือ ส่วนของ code ที่ใช้ในการควบคุมการทำงานต่างๆภายในเกม ตั้งเริ่มเกม จนจบเกม โดยในส่วน script ที่ใช้ในใน Unity จะถูกเขียนโดยภาษา C# เป็นหลัก ซึ่ง script จะถูกใช้งานได้โดยการรับ input ต่างๆจาก GameObject เช่น การกด w, a, s, d ในการสั่งให้ GameObject เคลื่อนที่ไปในทิศทางต่างๆ, การกด spacebar ในการสั่ง GameObject กระโดด เป็นต้น

2.2 พื้นฐาน Blender

Blender เป็น software ที่ใช้สำหรับสร้างงานทางสาย graphic 3D [2] สามารถสร้าง Model 3D ได้ ทำ texture ได้ รวมถึงสามารถทำ animation ได้ โดย Blender รองรับได้หลากหลายระบบปฏิบัติการ ไม่ว่าจะเป็น Windows, Mac OS, Linux แนวทางและโยชน์ในการประยุกต์ใช้งานโครงงานกับงานในด้านอื่นๆ

โดยส่วนต่างๆใน Blender ที่สำคัญ และช่วยในการสร้างเกม มีดังนี้

2.2.1 Workspaces

คือ หน้าต่างการทำงานต่างๆในโปรแกรม Blender ซึ่งในแต่ละหน้าจะทำหน้าที่แตกต่างกันไป โดยหน้าสำคัญๆมีดังนี้

1. Layout คือ หน้าหลักที่ใช้สำหรับออกแบบ และปั้น Model ต่างๆ
2. UV Editing คือ หน้าที่ใช้สำหรับการทำ UV texture ซึ่งใช้สำหรับการลงสี หรือพื้นผิวของ Model ซึ่งจะแสดงในรูปแบบภาพคลี่ของ Model

3. Shading คือ หน้าที่ใช้สำหรับควบคุมลักษณะของผิว Model เช่น ให้มีความมันวาว, มีความด้าน, ให้มีสีที่แตกต่าง เป็นต้น
4. Animation คือ หน้าที่ใช้สำหรับการออกแบบ สร้าง animation ต่างๆให้กับ Model เช่น ท่าทางการขยับตัว, การกระโดด, การกระพริบตา เป็นต้น

2.2.2 Mode

ใน Blender จะแบ่งการทำงานออกเป็น mode ซึ่งการทำงานของแต่ละ mode จุดประสงค์ และการทำงานที่แตกต่างกันออกไป โดยมี mode ที่สำคัญๆดังนี้

1. Object Mode คือ mode หลักที่ใช้สำหรับออกแบบ และปั้น Model ต่างๆ โดย mode นี้จะสร้าง object ต่างๆขึ้นมาได้ เช่น ทรงกลม, ทรงกระบอก, แผ่นระนาบ เป็นต้น
2. Weight Paint คือ mode ที่ใช้ในการควบคุม จัดการกับน้ำหนักของ Model โดยสามารถกำหนดน้ำหนักส่วนต่างๆของ Model ได้ตามที่ต้องการ ซึ่ง จะใช้ประโยชน์ตอนทำ animtion จะช่วยให้ animation ดูสมจริงมากยิ่งขึ้น
3. Texture Paint คือ mode ที่ใช้ในการลงสี texture โดยจะสามารถลงสีได้โดยตรงที่ตัว Model เลย
4. Edit Mode คือ mode ที่ใช้ในการจัดการกับ vertex, edge, face ของ Model ใช้สำหรับการต่อ-การดึง Model เช่น ใช้ในการสร้างแขน-ขาของ Model, ใช้สร้างของประดับตัว Model เป็นต้น
5. Sculpt Mode คือ mode ที่ใช้ในการปั้น Model ไม่ว่าจะเป็นการยืด การหด การทำให้ผิวเรียบเนียน

2.3 การเขียนโปรแกรมเชิงวัตถุ(Object Oriented Programming : OOP)

การเขียนโปรแกรมเชิงวัตถุ เป็นการเขียนโปรแกรมประเภทหนึ่ง โดยใช้แนวคิดในการพัฒนา software ที่มอง code เป็นวัตถุ(object)แทนการเขียน code เป็น streaming [4] ต่อกันยาวๆ การเขียนโปรแกรมเชิงวัตถุ ช่วยทำให้ Developer เห็นภาพรวมของ code ได้ง่ายขึ้น สามารถทำความเข้าใจ และแก้ไขข้อผิดพลาดต่างๆได้ถูกจุดอย่างรวดเร็ว เพราะการเขียน code แบบโปรแกรมเชิงวัตถุ code จะถูกแบ่งเป็นส่วนๆ(class)อย่างชัดเจน ซึ่งช่วยให้หา code ได้ง่ายขึ้น นอกจากนั้น หลักการการเขียนโปรแกรมเชิงวัตถุไม่ได้ยึดติดกับภาษาในการเขียนภาษาใดภาษาหนึ่ง ดังนั้นการเขียนโปรแกรมเชิงวัตถุ หรือ OOP จึงออกแบบมาเพื่อให้ code ที่เราเขียนมีแบบแผน เหมาะสมในการพัฒนา software ที่ซับซ้อน ซึ่งในการสร้างเกมก็ต้องใช้หลักการของ OOP ในการเขียน code เพื่อใช้ในการควบคุมการทำงานต่างๆภายในเกม หรือที่เรียกว่า script

ในการเขียนโปรแกรมเชิงวัตถุ เราจะเทียบ code กับวัตถุในชีวิตจริง เช่น มนุษย์(player) ซึ่งจะให้ player เป็นวัตถุ โดยสิ่งที่ player ต้องมีก็คือ คุณสมบัติ(Attribute) เช่น ผู้ชาย, สูง, ผมสั้น เป็นต้น และต้องมี พฤติกรรม(Behavior) เช่น เดิน, วิ่ง, กระโดด เป็นต้น การเขียนโปรแกรมก็ต้องทำให้ code ของเรามีคุณสมบัติและพฤติกรรมเช่นเดียวกันกับ player ซึ่งทำได้โดยการสร้าง class ของ player ขึ้นมา โดยใน class ที่สร้าง นั้นจะมีทั้ง Attribute และ Behavior ของ player ดังที่กล่าวมา และเมื่อจะนำ class ไปใช้งาน จะทำได้โดยการสร้าง object ขึ้นมา เปรียบเสมือนเป็น player หนึ่งคน โดย player ที่สร้างมานั้น จะมีทั้ง Attribute และ Behavior เหมือนของ class player ดังกล่าว

สำหรับการเขียนโปรแกรมเชิงวัตถุ หรือ OOP ประกอบไปด้วยหลักการที่สำคัญอยู่ 4 ข้อ ได้แก่

1. Encapsulation คือ การห่อหุ้มข้อมูล หรือการซ่อนข้อมูล สามารถทำได้โดยการกำหนดสถานะของ สิ่งที่ไม่อยากให้ใครเข้าถึง หรือแก้ไขได้ ให้มีสถานะ เป็น private กล่าวคือ เรามาสามารถกำหนดสถานะให้กับ Attribute และ Behavior ภายใน class ต่างๆให้เป็น private ได้ ซึ่งจะทำให้ class ภายนอก หรือผู้ใช้ ไม่สามารถเข้ามา แก้ไขข้อมูลนั้นๆได้ ช่วยแก้ไขเหตุการณ์ที่เมื่อเรามีหลาย object ที่มาจากหลายๆ class เมื่อไม่ได้กำหนดสถานะเป็น private จะทำให้ทุกๆ object สามารถเข้าไปแก้ไขเปลี่ยนแปลง Attribute และ Behavior ของ object ตัวอื่นๆได้อย่างอิสระ ซึ่งอาจจะทำให้เกิดความผิดพลาดในการทำงานได้ถ้าถูกเปลี่ยนแปลงคุณสมบัติโดยไม่มีการควบคุม
2. Abstraction คือ การทำให้ผู้ใช้หรือ ภายนอกรู้เท่าที่จำเป็น ซึ่ง Abstraction เป็นผลพลอยได้จาก Encapsulation เพราะเป็นการเลือกเปิดแค่บางสิ่งในบาง object ให้คนอื่นเห็น หรือกล่าวได้ว่า object ภายนอกสามารถเรียกใช้ Attribute หรือ Behavior ของ object ตัวที่เปิดให้เข้าถึงได้ โดยที่ object ที่มาใช้งานไม่รู้ว่าการทำงานเบื้องหลังทำอย่างไรบ้าง แต่รู้แค่ว่า ใช้ในทำอะไร
3. Inheritance คือ การทำเรามี class ที่จะสร้างขึ้นมาใหม่ แต่ class ใหม่นั้นมีความใกล้เคียงกับ class เก่าที่มีอยู่แล้ว แต่ไม่ใช่ตัวเดียวกัน หากเราสร้าง class ใหม่โดยสร้าง Attribute และ Behavior ขึ้นมาเองใหม่หมด จะทำให้เปิดความสิ้นเปลือง ดังนั้น Inheritance จึงมาช่วยแก้ไขในส่วนนี้ โดยการที่ class ใหม่ที่เราสร้าง จะสามารถนำ Attribute และ Behavior ของ class เก่ามาใช้ได้โดยไม่ต้องสร้างขึ้นมาใหม่ และสามารถแก้ไข เพิ่มลดการทำงานได้ โดยจะเรียกการแก้ไขแบบนี้ว่า การ Overdrive
4. Polymorphism คือ การที่เรามีหลายๆ class ที่คล้ายๆกัน และอยากให้แต่ละ class เหล่านั้นมี Attribute และ Behavior ที่เรียกใช้งานในทุกๆ class ให้ได้เหมือนๆกัน จึงใช้ Polymorphism มาช่วยแก้ไขปัญหาเหล่านี้ โดยการสร้าง class ต้นแบบที่มี Attribute และ Behavior ที่เราต้องการเตรียมไว้ และให้ class ที่คล้ายๆกันดังกล่าว เข้ามาสืบทอดไปสร้างการทำงาน ของตัวเองเอง เช่น class หมา กับ class แมว มี Behavior การเกิดเหมือนกัน เมื่อเราใช้ Polymorphism เราก็สร้างการทำงานของการเดินของแต่ละ class ได้ ซึ่งอาจจะเดินไม่เหมือนกัน แต่เราจะเรียกใช้ Behavior ได้ในลักษณะที่คล้ายกัน กล่าวคือ Behavior มีชื่อเดียวกัน ทำหน้าที่เดียวกัน แต่อาจจะมีขั้นตอนการทำงานคนละแบบกัน

2.4 Vector

ปริมาณทางคณิตศาสตร์มีสองแบบ คือ scalar เป็นปริมาณที่อธิบายด้วยปริมาณขนาด(magnitude)เพียงอย่างเดียว และ vector เป็นปริมาณที่อธิบายด้วยขนาด(magnitude) และทิศทาง(direction) ปริมาณทาง vector ถูกใช้ในหลากหลายศาสตร์ ไม่ว่าจะเป็น คณิตศาสตร์ ฟิสิกส์ เคมี และอื่นๆ ตัวอย่างปริมาณทาง vector เช่น ความเร็ว การกระจัด การเคลื่อนที่ต่างๆ แรง สนามแม่เหล็ก เป็นต้น ซึ่งในการสร้างเกม 3D ขึ้นมา ปริมาณทาง vector มีความสำคัญอย่างมาก เนื่องจากต้องใช้ในการคำนวณ ทิศทาง ตำแหน่งของ GameObject และใช้ในการทำให้ GameObject เกิดการ เคลื่อนที่ กล่าวคือ การสร้างเกม 3D ก็เหมือนการจำลองสร้างสภาพแวดล้อมให้เหมือนของจริง มีความเร็ว มีแรง มีมวล ซึ่งต้องอาศัยความรู้ทางปริมาณทาง Vector โดยความรู้ทางปริมาณทาง vector ที่ต้องใช้คือ

1. การเขียน vector โดย vector ประกอบไปด้วย 3 ส่วนคือ vector ในทิศทางแกน x y z ซึ่งจะนำมาเขียนในรูปแบบทางคณิตศาสตร์ได้โดยใช้ วงเล็บ เช่น (1,2,-4) ซึ่งก็คือ vector ที่ประกอบ ไปด้วย

vector ตามแกน x ที่มีขนาด 1 หน่วย vector ตามแกน y ที่มีขนาด 2 หน่วย และ vector ตามแกน z ที่มีขนาด -4 หน่วย

2. Magnitude vector คือ การหาขนาดของ vector โดยขนาดของ vector จะเป็นปริมาณทาง scalar ซึ่งใช้ในการคำนวณขนาดต่างๆได้ โดยสามารถหาขนาดของ vector ได้จาก สูตร vector V = (x, y, z) มีขนาดเท่ากับ

$$|\vec{V}| = \sqrt{x^2 + y^2 + z^2}$$

โดยสัญลักษณ์ขนาดของ vector V ใดๆแทนด้วย $|\vec{V}|$

3. Normalize vector คือ การทำให้ vector ที่เรามี มีขนาดเป็น 1 หน่วย หรือเรียกว่า vector 1 หน่วย โดยสามารถทำได้จากการทำ Normalization ซึ่งก็คือ การนำขนาดของ vector มาก หารกับ vector ตัวนั้นๆ ดังนี้

$$\hat{V} = \frac{\vec{V}}{|\vec{V}|}$$

เมื่อ $\vec{V}$ คือ vector ใดๆ และ $\hat{V}$ คือ vector 1 หน่วยของ $\vec{V}$

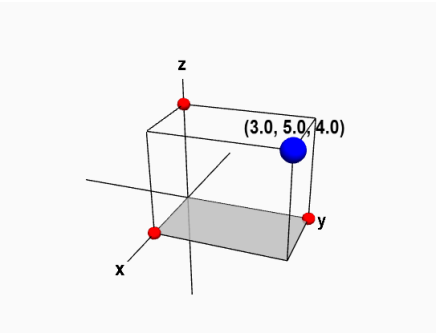
4. Distance คือ การหาระยะห่างระหว่าง vector 2 ตัวใดๆ ซึ่งเป็นปริมาณทาง scalar นำมาประยุกต์ใช้ในการคำนวณตำแหน่งต่างๆ, หาแรงที่ต้องใช้ได้ ซึ่งหาระยะทางได้จากสูตร

$$D_{UV} = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}$$

เมื่อ D_{UV} คือ ระยะห่างระหว่าง $\vec{U}$, $\vec{V}$ และ $\vec{U} = (u_x, u_y, u_z)$, $\vec{V} = (v_x, v_y, v_z)$

2.5 ระบบพิกัดคาร์ทีเซียนสามมิติ (Cartesian coordinate system)

เป็นระบบที่ใช้กำหนดตำแหน่งของจุดแต่ละจุดบนพิกัดฉาก โดยอ้างถึงตัวเลข 3 จำนวน ซึ่งแต่ละจำนวนเรียกว่า พิกัด x พิกัด y และพิกัด z ของจุดนั้นๆ และเพื่อที่จะกำหนดพิกัดของจุดใดๆ จะต้องมีเส้นแกนสามเส้นตัดกันเป็นมุมฉากที่จุดกำเนิด ได้แก่ แกน x แกน y และแกน z ซึ่งเส้นแกนดังกล่าวจะมีหน่วยบ่งบอกความยาวเป็นระยะ ระบบพิกัดคาร์ทีเซียนสามมิติยังสามารถใช้ได้ในปริภูมิสองมิติ หรือในมิติที่สูงกว่าได้ ตัวอย่างเช่น จุด a อยู่ที่พิกัด x = 3 พิกัด y = 5 พิกัด z = 4 หรือเขียนเป็น (3, 5, 4)

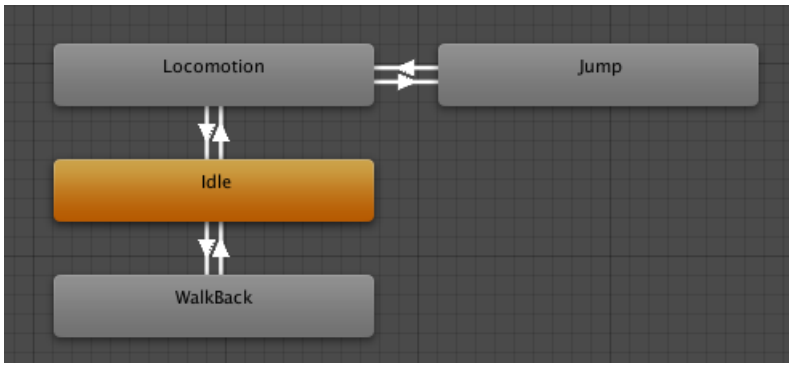


รูปที่ 2.1: ตัวอย่างจุดบนระบบพิกัดคาร์ทีเซียนสามมิติ

ซึ่งในการสร้างเกมเราจำเป็นต้องระบุตำแหน่งต่างๆของ GameObject ดังนั้นการสร้างเกมใน Unity จึงจำเป็นต้องรู้จักกับระบบพิกัดคาร์ทีเซียนสามมิติด้วย

2.6 State Machine

คือ การทำงานแบบมีการเปลี่ยนสถานะการทำงาน(state)ไปเรื่อยๆ เป็นขั้นตอนๆอย่างชัดเจน ซึ่งเป็นการมองสถานะการทำงานต่างๆ เป็น state โดยที่เมื่อ input เข้ามาก็จะมีการเปลี่ยนสถานะการทำงาน(เปลี่ยนไป state ใหม่) หรืออาจจะไม่เปลี่ยนก็(อยู่ที่ state เดิม)ได้ขึ้นกับ input ซึ่งการจะเปลี่ยนสถานะแต่ละครั้ง ขึ้นกับหลายๆอย่าง ได้แก่สถานะปัจจุบัน input เงื่อนไขต่างๆที่มีผลต่อการเปลี่ยนสถานะ ตัวอย่าง state machine เช่น ระบบประตูอัตโนมัติ โดยจะมีอยู่สอง state คือ ปิด และเปิด โดยเมื่อถ้าประอยู่ state ”ปิด” แล้วมี input คือ มีคนอยู่หน้าประตู หลังจากได้รับ input ประตูก็จะเปลี่ยน state ไปเป็น ”เปิด” ซึ่งถ้าไม่มีคนแล้วประตูก็จะเปลี่ยน state เป็น ”ปิด” ซึ่งในการสร้างเกมมีการนำ state machine เข้ามาใช้ในการแสดง animation ต่างๆ ไม่ว่าจะเป็น การเดิน การวิ่ง การกระโดด โจมตี และอื่นๆ



รูปที่ 2.2: ตัวอย่าง State Machine ที่ใช้ควบคุม animation

2.7 ภาษา C#

เป็นภาษาคอมพิวเตอร์ที่เขียนโปรแกรมแบบ multi-paradigm [3] โดยมีรูปแบบกฎเกณฑ์ และ ข้อบังคับในการเขียนที่เข้มงวด ซึ่งมีคุณสมบัติในการเขียนแบบ function เหมือนกับการเขียนภาษาทั่วไป และเป็นการเขียนโปรแกรมแบบ OOP โดย C# ถูกพัฒนาโดย Microsoft ภายใต้ .NET Framework [6] โดยในการพัฒนาภาษา C# นี้ มีความตั้งใจให้เป็นภาษาที่เขียนง่าย ทันสมัย

ใน Unity ภาษา C# ถูกนำมาใช้เป็นภาษาหลักในการเขียน script หรือ code ที่ทำหน้าที่จัดการกับการทำงานต่างๆในเกม โดยรองรับหลากหลาย Framework เช่น Visual studio, VS code เป็นต้น

2.8 ความรู้ตามหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงงาน

ในการทำโครงงานนี้กลุ่มของพวกเราได้นำความรู้ตามหลักสูตรต่างๆ มาประยุกต์ใช้ ซึ่งได้แก่

2.8.1 ความรู้ที่ได้จากหลักสูตรวิชา Object Oriented Programming 261200

1. การเขียนโปรแกรมเชิงวัตถุ(Object Oriented Programming : OOP) โดยการเขียนโปรแกรมเชิงวัตถุ กลุ่มของพวกเราได้นำมาใช้ในการวางแผน โดยการสร้าง interface และเขียน script ควบคุมการทำงาน โดยจะมีการออกแบบระบบต่างๆในเกมเป็น class เช่น class ของ characterController ใช้ควบคุมการควบคุมตัวละครโดยรวม, Status ใช้ในการควบคุมสถานะต่างๆของตัวละคร, Attack-Point ใช้คำนวณ damage จากการต่อสู้ เป็นต้น
2. การนำเอา design patterns มาใช้ในการเขียน code โดยในส่วนที่ต้องการให้ object ใดๆมีเพียงแค่ตัวเดียวได้มีการนำ Singleton มาใช้ เพื่อป้องการทำงานซ้ำซ้อนของ code, ในส่วน ที่เป็นการคำนวณ damage ซึ่งการ damage มีอยู่หลากหลายรูปแบบ เช่น damage จากการโจมตีแบบ 1 hit, damage จากการโจมตีแบบต่อเนื่อง(ติดสถานะต่างๆ), damage จากการโจมตี magic attack เป็นต้น ซึ่งเพื่อให้ code มีความยืดหยุ่นเมื่อมีการโจมตีรูปแบบใหม่เพิ่มเข้ามา จึงได้มีการใช้ Strategy เข้ามาช่วย ซึ่งจะทำการสร้าง interface รวมที่คอยจัดการเกี่ยวกับการคำนวณ damage ต่างๆไว้ที่เดียวกัน นอกจากนี้ยังมี code ส่วนอื่นๆก็มีการนำ design patterns มาใช้เช่นกัน

2.8.2 ความรู้ที่ได้จากหลักสูตรวิชา Calculus III 206261

1. การคำนวณทางปริมาณ vector โดยได้นำความรู้ต่างๆในเรื่อง calculus vector มาใช้ในการคำนวณต่างๆภายในเกม เช่น ทิศทางการเดิน, ทิศทางการโจมตี เป็นต้น รวมถึงการนำความรู้ทาง vector มาช่วยในการทำความเข้าใจการทำงานต่างๆของ method ต่างที่อยู่ใน Unity
2. ความเร็ว ความเร่ง โดยได้นำความรู้เรื่อง อัตราการเปลี่ยนแปลงของความเร็ว คือ ความเร่ง มาประยุกต์ใช้ในการหาความเร็วของวัตถุ

2.8.3 ความรู้ที่ได้จากหลักสูตรวิชา Data Structure 261217

การเก็บข้อมูลในรูปแบบต่างๆ โดยใน Unity การเขียน script บางครั้งต้องมีการเก็บข้อมูลต่างๆในเกมที่มีลักษณะที่แตกต่างกันออกไป ซึ่งข้อมูลเหล่านี้ จะเหมาะกับการเก็บใน structure ต่างๆแตกต่างกันออกไป เช่น การเก็บของที่ตัวละครเก็บได้ ควรจะเก็บเป็นรูปแบบ key และ value โดย key เป็น ชื่อของสิ่งของ ส่วน value คือ จำนวนของสิ่งของนั้นๆ เป็นต้น

2.8.4 ความรู้ที่ได้จากหลักสูตรวิชา ฟิสิกส์ 1 207105

1. ปริมาณทางฟิสิกส์ โดยในการสร้างเกม ระบบของเกมจะอิงตามหลักความเป็นจริง ซึ่งได้มีการใช้ปริมาณทางฟิสิกส์ต่างๆ เช่น ความเร็ว, ความเร่ง, แรงเสียดทาน, แรงโน้มถ่วง, มวล เป็นต้น
2. การเคลื่อนที่ต่างๆ มีการเคลื่อนที่หลากหลายรูปแบบที่สามารถเกิดขึ้นภายในเกมได้ เช่น การเคลื่อนที่ในแนวดิ่ง, การเคลื่อนที่วิถีโค้ง(projectile), การเคลื่อนที่แบบมีแรงเสียดทานต่างๆ เป็นต้น

2.8.5 ความรู้ที่ได้จากหลักสูตรวิชา discrete mathematics 261216

1. state machine โดยใช้ในการควบคุมการทำงานของ animation ของตัวละครต่างๆภายในเกม
2. ความรู้ทางด้านตรรกศาสตร์ ใช้ในการออกแบบ เงื่อนไขต่างๆภายในเกม

2.9 ความรู้นอกหลักสูตรซึ่งถูกนำมาใช้หรือบูรณาการในโครงงาน

ในการทำโครงงานนี้กลุ่มของพวกเราได้นำความรู้นอกหลักสูตรต่างๆ มาประยุกต์ใช้ ซึ่งได้แก่

2.9.1 ความรู้ทางด้านการสร้างเกมโดยใช้ Unity

เนื่องจากเกมของกลุ่มพวกเราใช้ Unity ในการพัฒนา ดังนั้นจึงได้มีการศึกษาการใช้งาน function ส่วนประกอบต่างๆใน Unity ดังที่กล่าวไว้ในข้อ 2.1 พื้นฐาน Unity และได้มีการไปศึกษาในส่วนของ VFX หรือ Visual effect เพื่อนำมาปรับใช้กับตัวเกมหลัก

2.9.2 ความรู้ทางด้านการเขียนภาษา C#

เนื่องจากเกมของกลุ่มพวกเราใช้ Unity ในการพัฒนา ซึ่งใช้ภาษา C# ในการเขียน script กลุ่มของพวกเราจึงได้ทำการศึกษาเพิ่มเติมกันเอง โดยอาศัยสื่อต่างๆทางอินเตอร์เน็ต เช่น youtube, unimy, google เป็นต้น ดังที่กล่าวไว้ในข้อ 2.7 ภาษา C#

2.9.3 ความรู้ทางด้านการปั้น Model โดยใช้ Blender

เนื่องจากมาจากมี asset บางอย่างที่จำเป็นต้องมีในเกม แต่ asset เหล่านั่นไม่สามารถหาซื้อได้ หรือมีราคาแพงจนเกินไป หรือไม่มีขาย จึงจำเป็นที่จะต้องปั้น Model ขึ้นมาใช้เอง เช่น ถังน้ำ, ก้อนหิน, บันได เป็นต้น

และในบางครั้ง Model ที่ซื้อมามี shader ที่ไม่ต้องการ หรืออยากปรับ shader ให้เป็นรูปแบบที่ต้องการ จึงต้องใช้ Blender ช่วยในการ จัดการกับประเด็นเหล่านี้ ดังที่กล่าวไว้ในข้อ 2.2 พื้นฐาน Blender

บทที่ 3
โครงสร้างและขั้นตอนการทำงาน

ในบทนี้จะกล่าวถึงการการออกแบบและพัฒนาเกมทั้งในส่วนของระบบเกม เนื้อเรื่อง เป้าหมายของเกม ระบบต่างๆที่มีให้ผู้เล่นใช้ในเกม และส่วนของเกมเพลย์ การต่อสู้กับศัตรู และวิธีการเล่นเกมเบื้องต้น เพื่อให้ผู้เล่นได้เข้าใจเกมมากยิ่งขึ้น

3.1 เนื้อเรื่องของเกม(Game story)

โลกของเกมนี้จะเป็นโลกแฟนตาซีมีการใช้เวทย์มนต์ ผู้เล่นจะได้รับบทเป็นเด็กน้อยที่มีพรสวรรค์ที่เกิดมาในรอบพันปี โดยโลกในปัจจุบันถูกจอมมาร ผู้ชั่วร้ายรุกรานอยู่ จึงต้องออกไปปราบจอมมารผู้ชั่วร้าย แต่ยังไม่มีความสามรถมากพอจึงต้องออกเดินทางเพื่อให้แข็งแกร่งขึ้นจนสามารถเอาชนะจอมมาร ได้และนำพาความสุขกลับมาสู่โลกอีกครั้ง โดยระหว่างการเดินทางผู้เล่นจะได้พบศัตรูหลากหลายรูปแบบซึ่งต้องใช้วิธีรับมือที่แตกต่างกัน และแก้ไขปริศนาต่างๆภายในเกม

3.2 User Interface (UI)

รูปที่ 3.1: ตัวอย่างที่ 1 UI ในเกมโดยอ้างอิงจากเกม genshin impact

รูปที่ 3.2: ตัวอย่างที่ 2 UI ในเกมโดยอ้างอิงจากเกม genshin impact

3.2.1 หน้าหลัก(Main Menu) เป็นหน้าแรกที่เข้าเกมมาแล้วเจอหน้าเมนูนี้มีตัวเลือกดังนี้

1. New Game คือเริ่มเกมใหม่ตั้งแต่ต้น
2. Continue เป็นการเล่นต่อจากที่เซฟไว้ล่าสุด
3. Options คือการตั้งค่าตัวเกม เช่น ปรับระดับเสียง
4. Exit คือการออกจากเกม

3.2.2 หน้าหยุดเกมชั่วคราว(Pause Menu) เมื่อเรากดปุ่มหยุดเกมชั่วคราวมีตัวเลือกดังนี้

1. Resume เล่นต่อ
2. Options เหมือนกับในหน้าแรก จะมีการตั้งค่าภายในเกม
3. Save and Exit เซฟเกมและกลับไปยังหน้าMain Menu

3.2.3 หน้าขณะเล่น(In Game UI) จะแสดงดังนี้

1. HP แสดงค่าพลังชีวิตของตัวละคร
2. Rage แสดงจำนวนค่าความโกรธที่สะสมไว้เมื่อครบจะใช้ ultimate skill ได้
3. Active Skills ที่ผู้เล่นจัดไว้ โดยจะแสดงว่าพร้อมใช้งานหรือว่าติดคูลดาวน์อยู่
4. Ultimate Skill ที่ผู้เล่นจัดไว้ โดยจะแสดงว่าพร้อมใช้งานหรือว่าติดคูลดาวน์อยู่
5. Character กดเพื่อเข้าไปยังหน้าต่างดู status ของผู้เล่น
6. Inventory กดเพื่อเข้าไปหน้ากระเป๋าเพื่อจัดการไอเทมต่างๆ และจัดอาวุธที่นำมาใช้งาน
7. Skills Menu กดเพื่อเข้าไปจัดสกิลเพื่อเอามาใช้งาน และปลดล็อคสกิลใหม่ๆ
8. Quest กดเพื่อเข้าไปดูเควสต่างๆที่รับมาทั้งเควสหลัก หรือเควสเสริม
9. Map แสดงจุดที่ผู้เล่นยืนอยู่ในแผนที่
10. Pause Button กดหยุดเกมชั่วคราว
11. Item shortcut ไอเทมที่เลือกไว้เพื่อกดใช้งานได้อย่างรวดเร็ว

3.2.4 หน้าพูดคุย(Talk UI) เมื่อมีการพูดคุยกับnpcหรือcut sceneที่มีการเล่าเนื้อเรื่อง

1. Text แสดงข้อความคำพูดและผู้พูด
2. Answers ตัวเลือกคำพูดที่ผู้เล่นจะเลือกตอบตามสถานการณ์

3.3 Game System

3.3.1 ระบบค่าสถานะ(Character Status)

ทั้งผู้เล่นและ monster จะมีค่าสถานะต่างๆที่ไม่เท่ากัน ขึ้นอยู่กับเลเวลและอุปกรณ์ที่สวมใส่อยู่ โดยค่าสถานะที่มีเฉพาะในผู้เล่นคือ Rage เป็นการสะสมความโกรธ มีค่าเพิ่มขึ้นเมื่อโจมตี monster สะสมค่าRageเพื่อใช้งานUltimate skill ส่วนค่าสถานะที่มีทั้งในผู้เล่นและ monster ได้แก่

1. Level เลเวลอัพแล้วจะมีค่าสถานะต่างๆสูงขึ้น และได้แต้มมาปลดล็อคสกิลของผู้เล่น
2. HP(พลังชีวิต)
3. Physical attack(พลังโจมตีกายภาพ)
4. Magic attack(พลังโจมตีเวท)
5. Physical Defence(พลังป้องกันกายภาพ)
6. Magic Defence(พลังป้องกันเวท)

3.3.2 ระบบสกิล(Skills System)

ผู้เล่นสามารถปลดล็อคได้ผ่าน skill tree ได้แต้มมาปลดล็อคเมื่อเลเวลอัพ แต่ละสกิลมีเงื่อนไขการปลดล็อคที่ไม่เหมือนกัน สามารถเลือกสกิลที่ปลดล็อคแล้วมาจัดใส่ช่องใช้งานมีให้3ช่อง มีปุ่มกดแต่ละช่อง Q E และ R โดยใส่ Active Skills ได้2ช่อง Q E และ Ultimate Skill ได้ 1 ช่อง R

3.3.3 ระบบไอเทม(Item System)

ไอเทมแบ่งออกเป็น อุปกรณ์สวมใส่ ไอเทมที่ต้องกดใช้งาน และทอง

1. อุปกรณ์สวมใส่: อุปกรณ์แต่ละชิ้นจะบวกค่าสถานะให้ตัวละครไม่เท่ากัน และค่าสถานะดังกล่าวจะขึ้นอยู่กับเลเวลของอุปกรณ์ โดยไม่สามารถอัพเลเวลได้ และสามารถได้รับจากการดรอปของหลังจาก monster ตาย โดยแบ่งออกเป็น อาวุธ, ชุด, แหวน (ชุดและแหวนจะไม่แสดงให้เห็นที่ตัวละคร)
2. ไอเทมกดใช้งาน: เมื่อกดใช้งานจำนวนของไอเทมก็จะลดจำนวนลง ได้แก่ ยาฟื้นฟูเลือด, บัพชั่วคราว เช่นบัพ damage, defence และมีช่อง item shortcut สามารถเลือกไอเทมกดใช้งานมาไว้ จะทำให้ใช้ไอเทมนั้นระหว่างต่อสู้ได้
3. ทอง: มีไว้สำหรับซื้อของที่ร้านค้า ซึ่งจะอยู่ในเมืองของมนุษย์ โดยร้านค้าจะขายอุปกรณ์สวมใส่ และไอเทมกดใช้งาน

3.3.4 ระบบอาวุธ(Weapon System)

อาวุธที่ผู้เล่นใช้ได้มี 3 ประเภท คือ ดาบ ธนู หนังสือเวท ซึ่งอาวุธแต่ละประเภทจะมีเอกลักษณ์เฉพาะตัว และมี status ที่เพิ่มให้ผู้สวมใส่แตกต่างกันออกไป โดยผู้เล่นสามารถเปลี่ยนอาวุธระหว่างต่อสู้ได้ โดยอาวุธดังกล่าวจะต้องถูกสวมใส่อยู่เท่านั้น อาวุธอื่นๆที่ไม่ได้สวมใส่ไว้ จะไม่สามารถเปลี่ยนระหว่างการต่อสู้ไม่ได้ ซึ่งสวมใส่อาวุธได้ไม่เกิน 3 ชิ้น

1. ดาบ(Sword) คำนวณ damage จาก Physical attack ตีระยะใกล้ แรงสุด
2. ธนู(Bow) คำนวณ damage จาก Physical attack ตีระยะไกล
3. หนังสือเวท(Grimoire) คำนวณ damage จาก Magic Attack ตีระยะไกลแบบ mini-aoe [1]

3.3.5 ระบบภารกิจ(Quest System)

แบ่งออกเป็นสองประเภทได้แก่เควสหลักและเควสเสริม

1. เควสหลักหรือเควสเนื้อเรื่อง: ผู้เล่นจะสามารถทำได้รอบเดียวต่อเควส เมื่อทำเควสแรกสำเร็จก็จะมีเควสต่อไปให้ทำเรื่อยๆ จนกว่าจะจบเนื้อเรื่องของเกม ในแต่ละเควสก็จะได้รับรางวัลต่างๆ ซึ่งแตกต่างกันในแต่ละเควส โดยจะมีการเล่าเนื้อเรื่องของเกมไปในตัว ในขณะทำเควสผู้เล่นจะไม่สามารถกดยกเลิกเควสได้
2. เควสรองหรือเควสเสริม: ผู้เล่นสามารถทำเควสได้หลายรอบ โดยจะเป็นการให้ผู้เล่นไปทำภารกิจต่างๆ เช่น ทำตามคำขอของเด็กน้อยให้ไปตามหาเพื่อนที่หลงในป่า เป็นต้น ในการทำเควสรอง เมื่อทำเควสสำเร็จ ผู้เล่นจะได้ค่าประสบการณ์ เงินหรือไอเทมเป็นรางวัล และผู้เล่นสามารถกดยกเลิกเควสได้

3.3.6 ระบบมอนสเตอร์(Monster System)

แบ่งออกเป็นสามประเภทได้แก่

1. Normal monsters : พบเจอได้ทั่วไปในแมพบริเวณนอกเมือง monster แต่ละพื้นที่จะมีเลเวล และค่าสถานะไม่เหมือนกันขึ้นกับสถานที่และอื่นๆ monster มีขอบเขต และเส้นทางการเดินของตัวเอง เมื่อ monster พบผู้เล่น monster จะทำการโจมตีใส่ผู้เล่น ถ้าผู้เล่นเดินหนีออกนอกเขตได้ก็จะเลิกไล่ตามแล้วกลับไปจุดของตัวเอง พร้อมกับรีเซ็ตเลือดตัวเองด้วย ซึ่งหาก monster ตายมันจะเกิดใหม่เมื่อผ่านไประยะเวลาหนึ่ง นานมากน้อยแค่ไหนขึ้นอยู่กับชนิด และเลเวลของ monster นั้นๆ
2. Zone's Boss : พบในแต่ละพื้นที่ของเกม โดยพบได้เพียงตัวเดียวต่อพื้นที่เท่านั้น ซึ่งทั้งเกมจะมีทั้งหมดสามตัว แต่ละตัวจะประจำตำแหน่งในพื้นที่ต่างๆของใครของมัน ได้แก่ พื้นที่ป่าไม้, ทะเลทราย และ ป่าหิมะ สามารถฆ่าได้ครั้งเดียว เมื่อฆ่าได้จะดรอปแหวนแห่งพลังที่เมื่อรวบรวมครบสามวง แหวนทั้งสามจะรวมกันกลายเป็นสุดยอดแหวน ซึ่งใช้ในการทำลายบาเรียของทวีปของจอมมาร ทำให้ผู้เล่นสามารถบุกเข้าไปต่อสู้ในทวีปของจอมมารได้
3. Demon King : มีเพียงตัวเดียวในเกม เป็นบอสสุดท้ายของเกม ถ้าผู้เล่นสามารถเอาชนะจอมมารได้เกมก็จะจบ Demon King มีจะความสามารถที่หลากหลาย พลังป้องกันสถานะต่างๆสูง ต้องใช้ทักษะการเล่น และตัวละครที่เลเวลค่อนข้างสูง

3.4 ตัวละคร

3.4.1 ผู้เล่น

ตัวละครหลักของเกมจะเป็นเด็กผู้หญิง ดังรูป

รูปที่ 3.3: ตัวอย่างตัวละครของผู้เล่น

3.4.2 Normal Monster

Monster ในเกมจะมีหลากหลายประเภท โดยแบ่งหลักเป็น สามารถบินได้กับไม่สามารถบิน ดังรูป

รูปที่ 3.4: ตัวอย่างที่ 1 Normal Monster

รูปที่ 3.5: ตัวอย่างที่ 2 Normal Monster

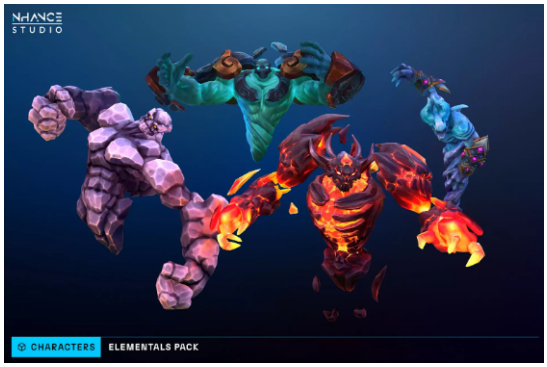
3.4.3 Zone's Boss

Zone's Boss ในเกมจะมีเพียง 3 ตัว ซึ่งมีรูปร่างแตกต่างกันไป

รูปที่ 3.6: ตัวอย่างที่ Zone's Boss

3.4.4 Demon King

Demon King ในเกมจะมีเพียง 1 ตัว เมื่อชนะได้ก็จะจบเกม



รูปที่ 3.7: ตัวอย่างที่ Demon King

3.5 Game play

3.5.1 การควบคุมตัวละคร

การควบคุมตัวละครในโครงงานนี้มีความสำคัญต่อความน่าสนใจ ความสนุก และการพลิกแพลง ภายในเกม เนื่องจากต้องใช้ความชำนาญ และไหวพริบในการควบคุม โดยสามารถอธิบายรายละเอียดการ ควบคุมตัวละครด้วยคีย์บอร์ดดังนี้

1. W A S D ใช้สำหรับการเดินหน้า ซ้าย ถอยหลัง และขวาตามลำดับ
2. Space bar ใช้ในการกระโดด
3. Mouse Left Click ใช้สำหรับโจมตีปกติ
4. Mouse Right Click ใช้สำหรับการโจมตีหนัก

5. Q, E ใช้งานActive Skills ที่เลือกไว้ 2 สกิล ตามลำดับ
6. R ใช้งานUltimate Skill
7. Left Alt ใช้เพื่อนำcursorออกมาจากการควบคุมมุมมองตัวละคร เพื่อนำมากดเมนูต่างๆ
8. Scroll Wheel ใช้เพื่อเปลี่ยนอาวุธในการต่อสู้
9. Left Shift ใช้เพื่อทำการพุ่งไปในทิศทางที่ควบคุมอยู่

3.5.2 ระบบการต่อสู้

มีการต่อสู้กับ monster ภายในเกม โดยเมื่อทำความเสียหายใส่ monster จนพลังชีวิตของมันหมด ผู้เล่นก็จะได้รับค่าประสบการณ์(Exp)และมีโอกาสสุ่มได้รับไอเทมที่มีเลเวลใกล้เคียงกับ monster นั้นและเงิน โดยถ้าเป็น zone's boss ก็จะดรอปไอเทมพิเศษที่จำเป็นสำหรับการทำเควสเนื้อเรื่องด้วย

3.5.3 เป้าหมายการเล่นเกม

ทำเควสเนื้อเรื่องไปเรื่อยๆระหว่างนั้นจะทำเควสรองหรือว่าสำรวจโลกเองก็ได้ โดยเควสเนื้อเรื่องจะให้ตัวหลักเดินทางไปยังพื้นที่ต่างๆ 3โซนคือป่า น้ำแข็ง ทะเลทราย เพื่อปราบมินิบอสของโซนนั้นและได้รับแหวนที่เพิ่มพลังผู้สวมใส่ เมื่อรวบรวมแหวนครบ3วง แหวนก็จะรวมกันเป็นสุดยอดแหวนที่สามารถทำลายบาเรียที่ปกป้องจอมมารอยู่ ทำให้ผู้เล่นสามารถโจมตีจอมมารได้ เมื่อชนะจอมมารได้ก็จะจบเกม

3.5.4 แผนที่

จะแบ่งออกเป็นสองทวีปโดย ฝั่งซ้ายเป็นที่อยู่ของมนุษย์ แบ่งเป็น 3 ภูมิภาคใหญ่ๆ คือ ป่า ทะเลทราย และป่าหิมะ มีเมืองหลวงอยู่ตรงกลาง ฝั่งขวาเป็นที่อยู่ของจอมมารและลูกน้องพื้นที่ส่วนใหญ่เป็นลาวาและฝุ่นควัน

บทที่ 4
การทดลองและผลลัพธ์

ในบทนี้จะเป็นการทดสอบระบบของตัวเกมว่ามีความสมบูรณ์มากน้อยเพียงใด โดยแบ่งหมวดหมู่เป็นความสมบูรณ์ของตัวละคร ความสมบูรณ์ของศัตรู แผนที่ การบรรลุเป้าหมายการเล่น และความสวยงามของเกม โดยรายละเอียดของแต่ละหมวดหมู่นั้นอธิบายได้ดังนี้

4.1 การทดสอบความสมบูรณ์ของตัวละคร

การทดสอบระบบการควบคุมตัวละคร จะทดสอบพื้นฐานคือการเดินหน้า ถอยหลัง ซ้าย ขวา กระโดด การเปลี่ยนอาวุธ การใช้Active SkillและUltimate Skill การหมุนมุมกล้องของตัวละครในแต่ละพื้นที่ของแมพไม่ให้มุมกล้องหลุดไปจากตัวละคร หรือมีบางส่วนของแผนที่บังมุมมองของผู้เล่น

4.2 ความสมบูรณ์ของศัตรู

ศัตรูในเกมจะต้องไม่ยากเกินไปจนทำให้ผู้เล่นไม่สามารถรับมือได้ หรือมีความบกพร่องที่ไม่สามารถทำให้ไปต่อได้ และไม่ง่ายเกินจนทำให้เกมน่าเบื่อไม่มีความท้าทาย

4.3 ความสมบูรณ์ของแผนที่

การหาจุดบกพร่องของแผนที่ในจุดต่างๆภายในเกม ไม่ให้มีจุดที่ตัวละครเดินไปแล้วเกิดหลุดออกจากแผนที่ไม่สามารถเดินไปในที่ที่ควรเดินไปได้ หรือเดินไปแล้วตัวละครติดในพื้นที่นั้นไม่สามารถนำตัวละครออกมาได้ อีกทั้งด้านความสวยงามของแผนที่ว่ามีอะไรที่แสดงผลออกมาไม่เหมือนที่ต้องการหรือไม่ จะทำการตรวจสอบโดยการนำตัวละครไปเดินในแต่ละจุด ที่คิดว่าอาจเกิดข้อบกพร่อง

4.4 การบรรลุเป้าหมายในการเล่น

ผู้เล่นต้องสามารถทำเควสหลักและรองได้โดยไม่เกิดปัญหาสามารถเดินทางไปยังพื้นที่ต่างๆ เมื่อจัดการบอสของพื้นที่แล้วต้องได้รับไอเทมที่นำไปใช้สู้กับจอมมาร และเมื่อชนะจอมมารเกมต้องจบลงและมีการกลับไปหน้าหลักเพื่อให้ผู้เล่นสามารถเริ่มเล่นใหม่อีกกี่รอบก็ได้เมื่อจบเกม

4.5 ความสวยงามของตัวเกม

ด้านความสวยงามก็เป็นส่วนสำคัญที่จะทำให้ตัวเกมดูน่าสนใจและน่าเล่นมากยิ่งขึ้น ไม่เพียงแค่ความ สวยงามภายในระบบเกมที่เราเล่น แต่รวมไปถึงหน้า UI ก็ส่งผลให้ผู้เล่นสนใจเล่นเกมของเรา ความสวยงามของสิ่งแวดล้อมในเกม เพลงประกอบที่ให้อารมณ์ความรู้สึกร่วม ตัวละครหลักที่มีความน่ารักสดใส มอนสเตอร์ที่มีความน่าเกรงขาม องค์ประกอบเหล่านี้จะทำให้ผู้เล่นสามารถสนุกไปกับเกมของเราได้มากขึ้น

บรรณานุกรม

[1] *AOE skill*. https://kochii.me/rov/rov-skill-aoe-

[2] *graphic 3D*. https://th.wikipedia.org/wiki/

[3] *Multi Paradigm*. https://th.wikipedia.org/wiki/

[4] *streaming coding*. https://www.jittagornp.me/blog/java-io-stream/.

[5] *visial effect(VFX)*. https://www.youtube.com/watch?v=WlzTZhDTaag.

[6] *Window, .NET Framework*. https://www.mindphp.com/