

Computer Science Department
Faculty of Informatics, Mahasarakham University

บทความวิจัย

میمเซอริวเวอร์: เซฟเดอะเวิลด์!

Meme Survivors: Save The World!

ณัฐพงศ์ สุทรวงศ์, พิริยะเดช นิลปะกะ, พรทิวา ปะวะระ

สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

63011212128@msu.ac.th, 63011212168@msu.ac.th, pornntiwa.p@msu.ac.th

บทคัดย่อ

โครงการปริญาานิพนธ์ฉบับนี้พัฒนาเกม “میمเซอริวเวอร์: เซฟเดอะเวิลด์!” ขึ้นมาซึ่งตัวเกมจะเป็นแนว Action Rouge-Like เอาชีวิตรอดจากฝูงมอนสเตอร์จำนวนมาก จุดเด่นของเกมแนวนี้คือการที่ถ้าตัวละครของผู้เล่นตายจะต้องเริ่มต้นใหม่ทั้งหมดทำให้ผู้เล่นจะต้องคิดวิธีเอาชีวิตรอดให้ดี และระบบการสู้มอาวุธ ไอเทมและสกิลที่จะทำการสู้ใหม่ทุกครั้งที่เล่นเกมจะสร้างประสบการณ์ที่แปลกใหม่ให้กับผู้เล่น โดยธีมหลักของเกมจะเป็นการนำ meme ต่าง ๆ ที่พบเห็นได้ตามโลกอินเทอร์เน็ตมาสร้างเป็นตัวละครผู้เล่นและมอนสเตอร์ภายในเกม รวมทั้งนำมาปรับแต่งให้เหมาะสมก่อนจะนำมาสร้างเป็นอาวุธ ไอเทมและสกิลต่าง ๆ ตัวเกมจะเป็นรูปแบบ 2D pixel เครื่องมือที่ใช้ในการพัฒนาจะเป็น Godot 3.5 ที่ภายในตัวโปรแกรมจะมีภาษาหลักของตัวเองคือ GD script

คำสำคัญ มีม (Meme), แอ็คชันโร้คไลค์ (Action Rouge-Like), Godot

1. บทนำ

ปัจจุบันผู้คนส่วนใหญ่ใช้เวลาว่างไปกับการเล่นเกมและโลกท่อนเทอร์เน็ต โดยเกมก็มีอยู่หลากหลายประเภทหลากหลายแนวสร้างความสนุกสนานเพลิดเพลินให้กับผู้เล่นและสร้างรายได้มาก มายให้กับผู้ผลิต มีทั้งแบบ 2 มิติ และแบบ 3 มิติ เช่น Simulator, Strategy, Role Playing Game, Sport, Fighting, Educate, Entertainment, First-Person Shooting, Music & Audio, Action Adventure, Social, Puzzle เป็นต้น และในขณะเดียวกันมีมซึ่งเป็นการเขียน การพูด ท่าทาง ภาพล้อเลียนที่มีความหมายเชิงตลกหรือปรากฏการณ์ลอกเลียนแบบที่เกิดขึ้นมามากมาย และสร้างความสนุกสนานในโลกอินเทอร์เน็ตจำนวนมากในช่วงหลายปีที่ผ่านมา

ในโครงการนี้ผู้พัฒนาได้สังเกตเห็นว่าเกมแนว Action Rouge-like กำลังได้รับความนิยมมากขึ้นเนื่องจากระบบการเล่นที่เข้าใจง่าย เพียงแค่กำจัดศัตรู เก็บ exp อัปเดตเลเวล สกิลหรืออาวุธที่จะสู้มาให้เลือกเรื่อย ๆ เนื่องจาก

เป็นการสุ่มเพื่อทำให้การเล่นในแต่ละรอบเกิดความหลากหลายจากการผสมรวม กันของอาวุธและสกิลทำให้เล่นได้ไม่เบื่อ ส่วนميمก็เป็นสิ่งที่เข้าถึงผู้คนได้ง่ายเต็มไปด้วยความตลกขบขันและความคิดสร้างสรรค์ สร้างเสียงหัวเราะในโลกอินเทอร์เน็ตได้ไว้มากมาย

ดังนั้นทางผู้จัดทำจึงขอเสนอเกม Meme Survivors: Save The World! ซึ่งเป็นแนวเกม Action Rougelike Survival ที่มีธีมเป็นميمต่าง ๆ ในอินเทอร์เน็ตที่ไม่ใช่แค่ความสนุกจากระบบการเล่นแต่ยังสามารถเพลิดเพลินไปกับميمต่าง ๆ ในเกมได้อีกด้วย

2. ทฤษฎีและระบบงานที่เกี่ยวข้อง

2.1.1 เกม

เกมเป็นกิจกรรมอย่างหนึ่งที่มอบความบันเทิงให้แก่มนุษย์หรือบางครั้งอาจใช้ประโยชน์ในด้านการศึกษาโดยมีโครงสร้างหลักประกอบด้วยเป้าหมาย และกฎกติกาที่ถูกกำหนดโดยมนุษย์สำหรับการ แข่งขันหรือพัฒนาทักษะด้านร่างกายการใช้พลังกำลังหรือความคิดเพื่อเอาชนะ ตัวอย่างเกมยุคแรกที่ยังคงได้รับความนิยมในยุคปัจจุบัน ได้แก่ ไพ่นกกระจอก ชักเย่อ หมากกรุก หมากล้อม เป็นต้น

2.1.2 เกม PC

เกม PC หมายถึงเกมที่เล่นบนเครื่องคอมพิวเตอร์ส่วนบุคคล เกมคอมพิวเตอร์ส่วนบุคคลเริ่ม แรกมีการพัฒนาให้มีรูปแบบการเล่นและกราฟิกที่เรียบง่าย ก่อนที่จะมีรูปแบบสลับซับซ้อนอย่างที่เราเห็นในปัจจุบันเกมคอมพิวเตอร์ส่วนบุคคลถูกผลิตขึ้นมาโดยผู้พัฒนาเกมหนึ่งคนหรือมากกว่า ซึ่งส่วนใหญ่มัก จะเป็นการรวมตัวกันของผู้เชี่ยวชาญหลายด้านที่รับผิดชอบในการดำเนินงานในตำแหน่งต่าง ๆ เช่น ตำแหน่ง Animator ที่รับหน้าที่ออกแบบการเคลื่อนไหวและฉากต่าง ๆ ตำแหน่ง Audio Engineer ที่รับผิดชอบเรื่องเสียงประกอบฉากของเกมหรือเสียงเอฟเฟคต่าง ๆ ที่เกิดขึ้นภายในเกม และตำแหน่ง Game Programmer รับผิดชอบการเขียนโค้ดให้กับเกม เป็นต้น

2.1.3 เกมแอ็คชั่น (Action Game)

เกมแอ็คชั่นเน้นการใช้ประสาทสัมผัสทั้งสายตาและมือในการควบคุมตัวละคร เพื่อให้ผ่านด่านภายในเกมหรือเพื่อที่จะทำให้ผ่านควอสที่เกมได้กำหนด เกมแอ็คชั่นบางเกมไม่ใช่ว่าผู้เล่นต้องเล่นทำควอส ตามด่านหรือต้องเอาชนะบอสให้ได้เพื่อจะผ่านด่านเท่านั้น แต่จะมีประเภทที่เกมจะให้เล่นไปเรื่อย ๆ ไม่มีจุดหมายที่แน่นอนว่าทำแบบนี้ผ่านหรือไม่ผ่านแต่จะให้ผู้เล่นเก็บไอเทมตามด่านหรือทำการ

กำจัดมอนสเตอร์เพื่อรับคะแนนในการเล่นในแต่ละรอบแทน แอ็คชั่นเป็นเหมือนคำเรียกภาพรวมของเกมประเภทต่าง ๆ เช่น เกมแนวยิงปืน ผจญภัย หรือ RPG ที่เป็นแนวเกมแยกย่อยออกมาอีกที ตัวอย่างเกม เช่น Bayonetta, Devil May Cry 5, Nier: Automata เป็นต้น

2.1.4 เกมแอ็คชั่นโร้คไลค์ (Action Rogue - Link)

เกมแอ็คชั่นโร้คไลค์คือเกมที่มีลักษณะสำคัญของการ Random หรือสุ่มไม่ว่าจะเป็นด่าน อาวุธ ไอเทม สกิล และศัตรูภายในเกมจะทำการสุ่มทุกครั้งที่คุณเริ่มเกมใหม่ทำให้ผู้เล่นต้องปรับตัวให้ยืดหยุ่น ตามไอเทมที่ได้รับมา เช่น รอบนี้อาจจะได้รับไอเทมที่เข้ากับด่าน ทำให้กำจัดมอนสเตอร์ได้ง่าย แต่รอบถัดไปได้รับไอเทมที่ไม่เข้ากับด่านที่กำลังเล่น ก็อาจจะทำให้การต่อสู้กับมอนสเตอร์ในรอบนั้นลำบากขึ้น อย่างไรก็ตามการที่ได้รับไอเทมที่ใช้งานยากไม่ได้แปลว่าจะทำให้ผ่านด่านในรอบนั้นไม่ได้เพราะมันขึ้นอยู่กับตัวผู้เล่นเองว่าจะสามารถปรับตัวให้เข้ากับอาวุธและด่านในรอบนั้นได้ดีแค่ไหน

ดังนั้น ความสนุกของมันคือ “การปรับตัว” ที่ผู้เล่นจะต้องพิจารณาว่า ตานี้เราสุ่มได้ไอเทมนี่มา เราจะเล่นยังไงแบบไหน ค่อย ๆ ไปตีไหม หรือว่าจะลุยแหลกเลยดี ตัวอย่างเกมเช่น Enter the Gungeon, Vampire

Survivors, The Binding of Isaac, Dead Cells เป็นต้น

2.1.5 มีม (Meme)

มีมเป็นรูปแบบของความคิดทางวัฒนธรรม สัญลักษณ์ หรือการปฏิบัติ ที่สามารถส่งผ่านจากจิตใจคนหนึ่งไปสู่อีกคนหนึ่ง ผ่านการเขียน การพูด ท่าทาง พิธีกรรม ภาพล้อเลียน และมีคำศัพท์ต่าง ๆ ในภาพที่มีความหมายเชิงตลกหรือปรากฏการณ์ลอกเลียนแบบอื่น ๆ คำว่า meme ในภาษาอังกฤษมาจากการผสมของคำว่า "gene" (ยีนหรือสิ่งสืบต่อพันธุกรรม) นักวิทยาศาสตร์ชาวอังกฤษ ริชาร์ด ดอว์กินส์ ได้คิดคำว่า "meme" ขึ้นมาในหนังสือ “The Selfish Gene (1976)” ในแนวคิดเกี่ยวกับการอภิปรายทฤษฎีวิวัฒนาการเกี่ยวกับการอธิบายการแพร่ของความคิดและปรากฏการณ์ทางวัฒนธรรมโดยรากศัพท์ของคำว่า meme มาจากศัพท์กรีก mimema (มีเมมา) ที่แปลว่าการลอกเลียน ในปี 2013 ดอว์กินส์ ได้กำหนดรูปแบบ Meme ทางอินเทอร์เน็ตว่าเป็นสิ่งที่เปลี่ยนแปลงโดยเจตนาโดยความคิดสร้างสรรค์ของมนุษย์ ซึ่งแตกต่างจากแนวคิดดั้งเดิมของเขาที่เกี่ยวข้องกับการกลายพันธุ์ ในเวลาต่อมา มีมถูกใช้เรียกแทนมุกตลก ภาพล้อเลียน ภาพทำซ้ำที่แพร่หลายออกมา

2.1.7 การค้นหาเส้นทาง (Pathfinding)

การทำงานโดยใช้การค้นหาแบบ Path Finding นั้น จะต้องใช้การคำนวณทางคณิตศาสตร์เข้าช่วย ซึ่งจะแตกต่างกันไปตามลักษณะของ Algorithm นั้น ๆ โดย Algorithm ส่วนใหญ่ที่ใช้ในการค้นหาแบบ Path Finding นั้นจะอธิบายด้วยการใช้กราฟที่มีการเชื่อมต่อเป็นโหนด ๆ โดยแต่ละโหนดจะมีเส้นทางเชื่อมไปหากัน การค้นหาเส้นทางเหล่านี้เพื่อให้ตัวละคร npc เข้ามาหาที่ตำแหน่งของผู้เล่น

2.1.8 การค้นหาแบบ A* (A Star Algorithm)

A* นั้นนิยมมากที่สุดสำหรับการค้นหาเส้นทาง เนื่องจากค่อนข้างยืดหยุ่นและสามารถใช้ได้ในบริบทที่หลากหลายโดยที่ A* นั้นมีลักษณะคล้าย Algorithm ของ Dijkstra ที่สามารถใช้ค้นหาเส้นทางที่สั้นที่สุดได้และมันเร็วพอ ๆ กับ Greedy Best-First-Search

เอสตาร์ใช้การค้นหาตามแนวกว้างและหาเส้นทางที่มีระยะทางน้อยที่สุดจากโหนดแรกไปสู่โหนดเป้าหมายซึ่งอาจจะมีได้หลายโหนด โดยใช้ฟังก์ชันฮิวริสติกแบบค่าของระยะทาง (ใช้สัญลักษณ์ $f(x)$) เพื่อที่จะหาลำดับการผ่านโหนดในกราฟ โดยฟังก์ชันดังกล่าวเป็นผลรวมของสองฟังก์ชันดังนี้

1. ค่าระยะทางทางจากโหนดเริ่มต้นมายังโหนดปัจจุบัน (ใช้สัญลักษณ์ $g(x)$)

2. ค่าประมาณฮิวริสติกที่ยอมรับได้ ของระยะทางที่จะถึงจุดหมาย (ใช้สัญลักษณ์ $h(x)$)

หลักการการทำงานของเอสตาร์คือ เมื่อเอสตาร์ท่องไปในกราฟ เอสตาร์จะเลือกเส้นทางที่มีค่าน้อยที่สุดที่มันทราบ โดยเก็บ Priority Queue ของเส้นทางอื่น ๆ ไว้ ถ้าระหว่างที่เอสตาร์ท่องไปแต่ละจุดแล้วเจอเส้นทางที่มีค่ามากกว่าเส้นทางอื่น ก็จะไปเลือกเส้นทางที่มีค่าน้อยกว่าใน Priority Queue แทน กระบวนการนี้จะทำต่อไปเรื่อย ๆ จนกระทั่งถึงจุดหมาย

2.2 ระบบงานที่เกี่ยวข้อง

2.2.1 Game Engine

Game Engine คือ โปรแกรมที่ใช้สำหรับสร้างเกมโดยตัวโปรแกรมจะมีสิ่งอำนวยความสะดวกให้กับนักพัฒนามากมาย อย่างการ Rander ภาพ 2D หรือการสร้างโมเดล 3D สามารถนำภาพ 2D มาสร้างเป็น Animation ตัวละครของเกม มีระบบฟิสิกส์ให้สิ่งของภายในเกม มีระบบการเล่นเสียงเพลงพื้นหลังหรือเอฟเฟคประกอบกรกระทำต่าง ๆ และสามารถสร้างเป็นฉาก 2D ต่าง ๆ จาก Tilemap กำหนดองค์ประกอบต่าง ๆ ภายในเกม ตัวอย่าง game engine ที่ใช้ในการพัฒนาเกม Godot, Armory, Unreal Engine, Cry Engine,

Defold, Corona, Unity, Frostbite Engine, RE Engine, Source Engine เป็นต้น

โดย engine แต่ละตัวก็มีความสามารถแตกต่างกันไปเช่นตัว engine ที่นิยมนำมาสร้างเกม 3D เป็นหลักอย่าง Armory แต่มีข้อจำกัดในการใช้งานเพราะต้องมีความชำนาญระดับหนึ่งถึงจะสามารถใช้งานได้อย่างมีประสิทธิภาพ และหากต้องการที่จะสร้างเกมที่มีความเสมือนจริง Cry Engine ก็เป็นตัวเลือกหนึ่งที่มีความสามารถมากพอจะสร้างเกมเสมือนจริงที่มีความคมชัดของภาพในระดับสูงและกำหนดรายละเอียดของภาพได้ดีอีกหนึ่งตัว หรือแม้แต่ว่า engine ที่ดัง ๆ อย่าง Unreal Engine ที่เหมาะกับทั้งมือใหม่และคนที่ชำนาญแล้วสามารถพัฒนาได้ทั้งเกม 2D และ 3D ลงรับการพัฒนาได้ตั้งแต่เกม 2D เบา ๆ ไปจนถึงเกม 3D กราฟฟิกสูง และ engine อีกตัวคือ Unity ที่ใช้งานง่ายทำได้ทั้งเกม 2D และ 3D มี package ให้มากมายสามารถเลือกนำไปพัฒนาตามแต่เกมที่เรากำลังสร้าง และ engine ที่สามารถทำได้ทั้ง 2D และ 3D อีกตัวคือ Godot ถึงแม้ว่า engine จะมีข้อจำกัดและไม่ได้ดังมากเท่า 2 engine ข้างบนแต่ก็เป็นอีก engine หนึ่งที่ใช้งานง่ายและใช้พื้นที่น้อยสามารถพัฒนาเกมที่ไม่ได้มีองค์ประกอบกว้างมากได้ออกมาดีไม่ต่างกัน

2.2.2 Godot

Godot (ออกเสียง Go-doh) สามารถทำงานได้หลายแพลตฟอร์ม (Cross platform) เช่น Window, Mac OS เป็นต้น และเป็นโปรแกรม open source ใช้งานฟรี ภายใต้สัญญา MIT ซึ่งไม่มีพันธะข้อผูกมัดหรือค่าสิทธิ (Royalty) ในการใช้งานแต่อย่างใด ผู้ใช้งานจะเป็นเจ้าของเกมที่ตนเองสร้างตลอดจนโค้ดทั้งหมดของตัวเอนจินโดยสมบูรณ์ เดิมทีถูกพัฒนาอยู่ที่ Argentina โดยนักพัฒนาซอฟต์แวร์ Juan Linietsky และ Ariel Manzur ในปี 2007 ให้กับบริษัทที่อยู่แถว Latin America ก่อนที่จะทำเป็นโปรแกรมฟรีให้คนทั่วไปใช้งาน ในปี 2014 Godot ได้เข้าร่วมกับ Software Freedom Conservancy ซึ่งเป็นองค์กรไม่แสวงหาผลกำไร ตัวโปรแกรมถูกสร้างมาเพื่อพัฒนาเกม 2D และ 3D ที่เกมสามารถรองรับการรันบน PC, Mobile และ web ได้ นอกจากพัฒนาเกมแล้ว ตัวโปรแกรมยังสามารถทำอนิเมชันได้ด้วย ภาษาโปรแกรมที่ Godot รองรับมี C#, GDscript, Visualscript, C++, C การเขียน script ให้กับเกมที่นิยมจะใช้ 2 ภาษาคือ C# และ Gdscript

2.2.3 GDScript

GDScript เป็นภาษาการเขียนโปรแกรมระดับสูงที่มีการเขียนแบบไดนามิก ใช้ไวยากรณ์ที่มีลักษณะคล้ายกับ Python โดยที่

โค้ดถูกเยื้องแทนที่จะใช้วงเล็บและเซมิโคลอน ทำให้ง่ายต่อการเรียนรู้ โค้ดส่วนใหญ่สามารถเขียนและเปลี่ยนแปลงได้อย่างรวดเร็วและไม่ยุ่งยาก ทำให้เขียนโค้ดน้อย อ่านโค้ดได้ง่ายขึ้น และไม่จำเป็นต้องคอมไพล์เพื่อทดสอบ ดังตัวอย่าง

```
func _ready():
    print("Hello World")
```

3. ขั้นตอนการดำเนินงาน

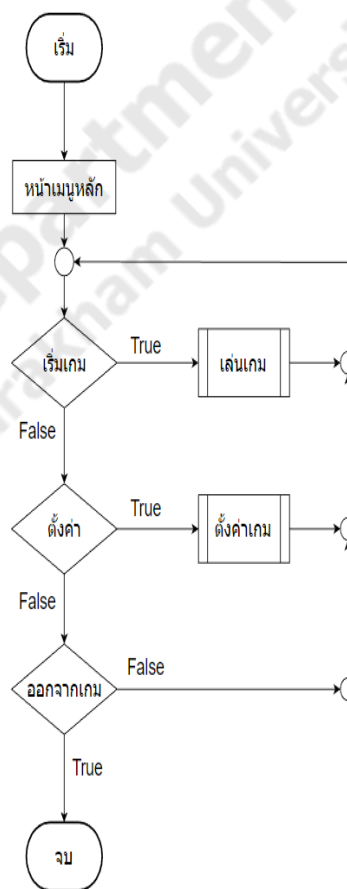
3.1 ระบบของเกม

3.2.1 ภาพรวมระบบการเล่นเกม

- (1) เกมเป็นแนว Action Roguelike มุมมองบุคคลที่ 3
- (2) มีตัวละครให้เลือกเล่น 3 ตัว และมีสกิลพิเศษที่แตกต่างกัน
- (3) มีมอนสเตอร์ธรรมดา 8 ชนิด และมอนสเตอร์พิเศษ 4 ชนิด
- (4) มีอาวุธ ไอเทมและสกิลทั้งหมดอย่างละ 6 ชนิด
- (5) ผู้เล่นสามารถครอบครองอาวุธและไอเทมได้อย่างละ 4 ชนิดต่อรอบ
- (6) โหมดปกติ เงื่อนไขการผ่านคือเอาชนะบอสในนาทิตี่ 20
- (7) โหมดพิเศษ เล่นต่อไปเรื่อย ๆ จนกว่าตัวละครผู้เล่นจะตาย

(8) เกมสามารถทำลายสิ่งของภายในแมพเพื่อครอบไอเทมพิเศษได้

(9) เกมสามารถนำเงินที่ได้จากการเล่นในแต่ละรอบมาอัพเพิ่มค่าสเตตัสให้กับตัวละครได้



ภาพประกอบที่ 1 ภาพรวมระบบ

3.3 การดำเนินเกม

3.3.1 เนื้อเรื่อง

ในวันหนึ่งโลกได้เผชิญหน้ากับหายนะที่คาดไม่ถึง เมื่อมีไวรัสคอมพิวเตอร์ตัวหนึ่งได้กลายพันธุ์แล้วเปลี่ยนให้ผู้คนในโลกจริงกลายเป็นมอนสเตอร์ที่มีมิติเป็นต้นแบบมัน

แพร่กระจายไปทั่วโลกอย่างรวดเร็วและทำให้
โลกล่มสลาย เหล่าผู้ที่มีภูมิคุ้มกันแม้จะกลาย
ร่างไปแล้วแต่ก็ยังคงสติได้อยู่จึงต้องหาทางเอา
ชีวิตรอดพร้อมกับช่วยโลกใบนี้ไว้

3.3.2 การควบคุม

ตารางที่ 1 ควบคุมตัวละคร

ปุ่มกด	คำอธิบาย
W A S D	W เดินขึ้นบน, A เดินไปซ้าย , S เดินไปขวา, D เดินลงล่าง
Z	ใช้สกิลพิเศษ
Esc.	เมนู

3.3.3 การคำนวณเงินที่ได้จากการเล่น

$$X = (((Y + P) + B) / 1,000) + L \quad (1)$$

X คือ จำนวนเงินที่จะได้รับจากการ
เล่นในแต่ละรอบ

Y คือ คะแนนที่ได้จากการกำจัดมอน
สเตอร์ (คะแนนที่ได้ขึ้นอยู่กับชนิดของมอน
สเตอร์)

P คือ คะแนนที่คิดจากเลเวลของผู้
เล่น (1 เลเวล = 2,000 คะแนน)

B คือ โบนัสคะแนนที่จะได้เมื่อชนะ
บอสในนาทิตี่ 20 ได้ (30%)

L คือ โบนัสเงินที่คิดจากการเลือก
เล่นในแต่ละระดับความยาก (0%, 30%,
50%)

3.4 Game Design

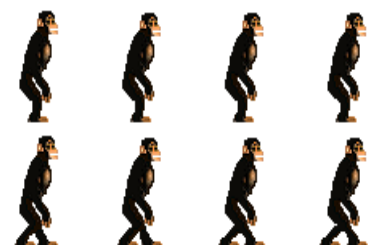
3.4.1 ตัวละคร



ภาพประกอบที่ 2 Doge



ภาพประกอบที่ 3 Pepe the frog



ภาพประกอบที่ 4 Monkey Caesar

3.4.2 การสร้างตัวละคร



ภาพประกอบที่ 5 สร้าง assets ตัวละคร

3.5 Json Design

```

1  {
2  "Players": {
3    "The_Doge": {
4      "Max_HP": 50,
5      "HP": 50,
6      "ATK": 15,
7      "SPD": 50,
8      "Pickup_Radius": 20,
9      "Spacial_Skill": true,
10     "Buy_Spacial_Skill": true,
11     "Collected_Coin": 10
12   },
13   "Monkey_Caesar": {
14     "Max_HP": 40,
15     "HP": 40,
16     "ATK": 20,
17     "SPD": 50,
18     "Pickup_Radius": 20,
19     "Spacial_Skill": true,
20     "Buy_Spacial_Skill": true,
21     "Collected_Coin": 10
22   },
23   "Pepe_The_Frog": {
24     "Max_HP": 50,
25     "HP": 50,
26     "ATK": 10,
27     "SPD": 60,
28     "Pickup_Radius": 20,
29     "Spacial_Skill": true,
30     "Buy_Spacial_Skill": false,
31     "Collected_Coin": 0
32   }
33 }
34 }

```

ภาพประกอบที่ 6 data ตัวละคร

```

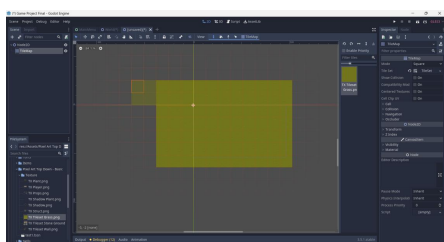
1  {
2    "Coin": {
3      "Num_Coin": 0
4    }
5  }

```

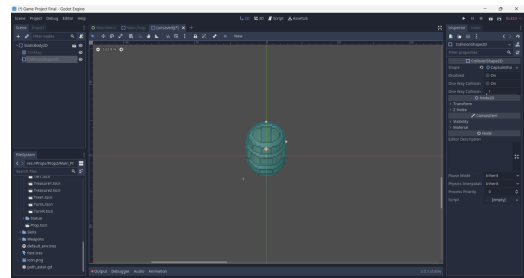
ภาพประกอบที่ 7 data เงิน

3.6 กระบวนการทำงานของระบบ

3.6.2 แมพและสิ่งของภายในแมพ



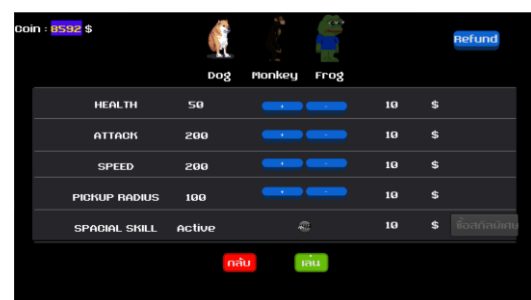
ภาพประกอบที่ 8 วาดแมพ



ภาพประกอบที่ 9 สร้างสิ่งของ

4. การทดสอบระบบ

การพัฒนาเกม “มีมเซอร์ไวเวอร์: เซฟเดอะเวิลด์!” ในส่วนของผู้เล่นจะได้รับประสบการณ์ในการเล่นแนว roguelike เอาชีวิตรอด ที่ผู้เล่นจะต้องคิดวางแผนการเล่นในแต่ละรอบว่าในแต่ละรอบที่สุ่มได้อาวุธหรือไอเทมชิ้นนั้นควรจะเล่นไปในทิศทางไหนดี อีกทั้งการที่มอนสเตอร์จะเดินเข้ามาหาผู้เล่นอยู่ตลอดเวลาทุกทิศทางจะทำให้ผู้เล่นมี Action กับเกมอยู่ตลอดเวลาทำให้ไม่เบื่อ



ภาพประกอบที่ 10 หน้าเลือกตัวละคร



ภาพประกอบที่ 11 ตัวอย่างเกม



ภาพประกอบที่ 12 ไอเทมพิเศษทำงาน

5. สรุปผลและข้อเสนอแนะ

5.1 สรุปผลและอภิปรายผล

การพัฒนาเกม “มีมเซอร์ไวเวอร์: เซฟเดอะเวิลด์!” ในส่วนของผู้เล่นได้รับประสบการณ์ใหม่ในการเล่นเกมแนว Roguelike ที่ต้องรู้จักการใส่ใจในตัวละครที่เล่นในแต่ละรอบเพราะการตายในเกมจะทำให้ต้องได้เริ่มต้นใหม่ โดยสิ่งที่จะดึงดูดให้ผู้เล่นเล่นเกมนี้คือการใช้อาวุธ ไอเทม และสกิลกำจัดมอนสเตอร์ที่มีจำนวนมากจะทำให้เกิดความพึงพอใจได้จากการมองดูมอนสเตอร์ที่โดนโจมตีค่อย ๆ หายไป และระบบที่ทำการสุ่มอาวุธ ไอเทมและสกิลพร้อมทั้งสิ่งของต่าง ๆ ใหม่ทั้งหมดจะทำให้ผู้เล่นมีความรู้สึกแปลกใหม่ทุกครั้งที่การเล่นแต่ละรอบ พร้อมทั้งระบบการโจมตีของอาวุธและไอเทมจะเป็นแบบอัตโนมัติทั้งหมดทำให้ผู้เล่นไม่จำเป็นต้องทำความเข้าใจอะไรมากสามารถเล่นแบบเพลิดเพลินเวลาว่างได้เลย และระบบควมมอนสเตอร์ที่ตั้งค่าให้เดินเข้าหาผู้เล่นอยู่ตลอดเวลาจะทำให้ผู้เล่นจดจ่ออยู่กับทิศทางในการเดินยังไงไม่ให้โดนมอนสเตอร์

พวกนั้นล้อมได้ ด้วยการจذبนี้จะทำให้ผู้เล่นใช้เวลาอยู่กับเกมนี้นานขึ้น

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

(1) แหล่งเรียนรู้ภาษาไทยมีน้อยการอธิบายการทำงานบางส่วนในเอกสารค่อนข้างเข้าใจยากและการสืบค้นด้วยภาษาอังกฤษเกิดความสับสนบ่อยครั้ง

(2) ปัญหาเกี่ยวกับชินที่ทำการสืบทอดมาจากชินหลัก ก็คือเมื่อทำการเปลี่ยนแปลงในชินลูกจะทำให้เกิดการเปลี่ยนแปลงในชินหลักด้วย ทำให้เกิดข้อผิดพลาดตามมาจากชินอื่น ๆ ที่ทำการสืบทอดเหมือนกัน

(3) ต้องเรียนรู้การทำแอนิเมชันภาพพิกเซลด้วยตัวเองตั้งแต่ต้น ทำให้เสียเวลาในการศึกษาค่อนข้างมาก และการเลือกตัวของโปรแกรมในการสร้างก็ต้องทำการศึกษาตัวโปรแกรมด้วย

(4) ปัญหาของเวอร์ชันของโปรแกรม Godot ที่ปัจจุบันเป็นเวอร์ชัน 4.0 แต่ตอนที่ผู้ทำได้เริ่มทำจะเป็นเวอร์ชัน 3.5 เมื่อทำการอัปเดตเป็น 4.0 ทำให้คำสั่งบางอย่างผิดแปลกไปหรือไม่สามารถใช้งานได้ทำให้ผู้จัดทำต้องอัปเดตโปรแกรมย้อนกลับไปเป็นเวอร์ชัน 3.5 เดิมจึงจะสามารถจัดการกับปัญหาได้ โดยแลกมากับตัวเวอร์ชัน 3.5 จะมีบางส่วนที่ทำงานแย่กว่า 4.0 และมี function บางส่วนที่ใช้ได้กับ

4.0 เท่านั้นทำให้ผู้จัดทำที่ใช้เวอร์ชัน 3.5 จึงจำเป็นต้องสร้าง function ที่ไม่มีขึ้นมาเอง

5.3 ข้อเสนอแนะ

(1) เพิ่มอาวุธ ไอเทมและสกิลเข้ามาเพิ่มเพื่อความหลากหลายในการเล่น

(2) สร้างอาวุธ และไอเทมที่เมื่อเลเวลสูงสุดแล้วจะเปลี่ยนรูปร่างและเปลี่ยนการโจมตีที่อลังการขึ้นเพื่อเพิ่มความน่าตื่นเต้นในการเปลี่ยนแปลง

(3) เพิ่มอีเว้นท์ที่เกิดจากมอนสเตอร์เพื่อความหลากหลายในการต่อสู้

(4) เพิ่มมอนสเตอร์และแมพใหม่ให้เกิดความหลากหลายในการเลือกด้านต่าง ๆ

6. อ้างอิง

1. ChaV. (2563). สร้างภาพพิกเซลและอนิเมชันง่าย ๆ ด้วย Piskel. สืบค้นเมื่อ 5 สิงหาคม 2565, จาก <https://yong-christmas.medium.com/สร้างภาพพิกเซลและอนิเมชันง่ายๆด้วย-piskel-dec95a64cd41>
2. Godot Engine community. (2565). Godot Docs – 3.5 branch. สืบค้นเมื่อ 5 สิงหาคม 2565, จาก <https://docs.godotengine.org/en/3.5/index.html>
3. Know Your Meme. (2566). รูปมีมที่

ใช้ทำตัวละคร อาวุธ สกิลและไอเทม. สืบค้นเมื่อ 5 สิงหาคม 2565, จาก <https://knowyourmeme.com/>

4. Wikipedia. (2565). การค้นหาแบบเอสตาร์. สืบค้นเมื่อ 28 กันยายน 2565, จาก <https://th.wikipedia.org/wiki/การค้นหาแบบเอสตาร์>
5. Wikipedia. (2566). Meme. สืบค้นเมื่อ 5 สิงหาคม 2565, จาก https://en.wikipedia.org/wiki/Meme#cite_note-Wired20130620-63
6. Wikipedia. (2566). อนิเมชัน. สืบค้นเมื่อ 8 สิงหาคม 2565, จาก <https://th.wikipedia.org/wiki/%E0%B9%81%E0%B8%AD%E0%B8%99%E0%B8%B4%E0%B9%80%E0%B8%A1%E0%B8%8A%E0%B8%B1%E0%B8%99>