

เกมวางแผนการรบโดยการเขียนโปรแกรมคำสั่ง
STRATEGY GAME WITH INSTRUCTION WRITING



ยอดเยี่ยม อัครานูวัฒน์
วชิรวิทย์ สุนทรวรจันทร

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เกมวางแผนการรบโดยการเขียนโปรแกรมคำสั่ง
STRATEGY GAME WITH INSTRUCTION WRITING



ยอดเยี่ยม อัทธานุวัฒน์
วชิรวิทย์ สุนทรวรจันทร์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2561

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เกมวางแผนการรบโดยการเขียนโปรแกรมคำสั่ง

STRATEGY GAME WITH INSTRUCTION WRITING

ผู้จัดทำ

1. นายยอดเยี่ยม อัครานวุฒินันท์ รหัสนักศึกษา 58011031

2. นายวชิรวิทย์ สุนทรวรจันทร รหัสนักศึกษา 58011086



อาจารย์ที่ปรึกษา

(อาจารย์วิบูลย์ พร้อมพานิชย์)

อาจารย์ที่ปรึกษาร่วม

(ผศ. ธนา หงษ์สุวรรณ)

เกมวางแผนการรบโดยการเขียนโปรแกรมคำสั่ง

นายยอดเยี่ยม อัครานูวัฒน์ 58011031

นายชริวิทย์ สุนทรวรจันทร์ 58011086

อาจารย์วิบูลย์ พร้อมพานิชย์ อาจารย์ที่ปรึกษา

ผศ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2561

บทคัดย่อ

ในปัจจุบัน เกมเป็นสื่อที่มีผู้คนให้ความสนใจเป็นจำนวนมาก ทำให้ก่อเกิดเป็นอาชีพต่างๆ ทำงานเกี่ยวกับเกม หนึ่งในนั้นคือนักพัฒนาเกม ซึ่งมีความสามารถในการออกแบบและพัฒนาเกม ออกมาให้มีคุณภาพและสร้างความบันเทิงให้แก่ผู้เล่น และทางผู้จัดทำได้มีความสนใจในการศึกษา และพัฒนาเกม ก่อให้เกิดโครงการที่ว่าด้วยการออกแบบและพัฒนาเกมประเภทวางแผนกลยุทธ์ โดยได้มีการผสมผสานการเขียน โปรแกรมคำสั่งในรูปแบบบล็อกไปในเกมการเล่น ซึ่งได้แรงบันดาลใจมาจาก blockly game เพื่อสร้างความแปลกใหม่ ความน่าสนใจ และความสนุกสนาน เพลิดเพลิน รวมไปถึงเป็นการฝึกฝนและพัฒนาการวางแผนและจัดการให้แก่ผู้เล่น สามารถเล่นได้ ในระบบปฏิบัติการไมโครซอฟท์วินโดวส์ โดยใช้โปรแกรม Unity engine เป็น โปรแกรมหลักในการพัฒนาเกม

Strategy Game With Instruction Writing

Mr. Yodyeam	Autanuwat	58011031
Mr. Wachirawit	Sunthonoaurachan	58011086
Mr. Wiboon	Promphanich	Advisor
Asst. Prof. Thana	Hongsuwan	Co-Advisor

Academic Year 2018

ABSTRACT

Nowadays, game is a media that has attracted a lot of people and make a career that work on game, one of them is game developer. Game developer have ability to design and develop games for quality and entertainment for players, and the organizers are interested in studying and developing the game, so we decided to do the project that about design and development of strategy game. There is a combination of instruction commands in the form of block in gameplay. Inspired by the blockly game to create novelty, interest and fun. It includes training and development, planning and management to the players. This game can play in Microsoft Windows operating system, by using the Unity engine is the main program for game development.

กิตติกรรมประกาศ

ปริญญาโทฉบับนี้สามารถจัดทำขึ้นมาได้อย่างสำเร็จลุล่วงได้ ต้องขอขอบคุณอาจารย์วิบูลย์ พร้อมพานิชย์ อาจารย์ที่ปรึกษาและ ผศ. ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษาร่วม ที่คอยรับคำปรึกษาและให้คำแนะนำต่างๆ เอาใจใส่ดูแล สอบถามความคืบหน้า และให้ความช่วยเหลือในทุกๆ เรื่องเป็นอย่างดี

ขอขอบคุณอาจารย์ทุกท่านได้อบรมและสั่งสอนให้วิชาความรู้ต่างๆ ทำให้ทางคณะผู้จัดทำได้นำความรู้ต่างๆ มาประยุกต์ใช้ในการจัดทำปริญญาโทฉบับนี้ รวมถึงความช่วยเหลือในด้านต่างๆ อีกมากมาย

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทั้งในภาควิชาวิศวกรรมคอมพิวเตอร์ หรือภาควิชาอื่นๆ ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้รับฟังคำปรึกษาและให้คำแนะนำต่างๆ รวมถึงขอขอบคุณห้องปฏิบัติการต่างๆ ของภาควิชาวิศวกรรมคอมพิวเตอร์และหอสมุดกลางของสถาบันที่ใช้เป็นสถานที่ทำงานในการทำปริญญาโทฉบับนี้

และท้ายที่สุด ขอขอบคุณผู้ปกครองของคณะผู้จัดทำที่คอยให้ความช่วยเหลือต่างๆ รวมถึงผู้ที่มีส่วนเกี่ยวข้องที่ไม่ได้กล่าวถึงทุกท่าน

ยอดเยี่ยม อัครานูวัฒน์
วิหริวิทย์ สุนทรวรจันทร์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 แผนการดำเนินงาน	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 เกม Strategy	3
2.2 Blockly game	4
2.3 Unity 3D	5
2.4 Fullerton's game prototype usability quality criteria	6
2.5 Game Development Life Cycle	7

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบและการพัฒนา	10
3.1 ระบบการเล่นหลัก	10
3.2 ข้อมูลรายละเอียดของไอเทมภายในเกม	14
3.3 เนื้อเรื่องของเกม	17
3.4 เกมการเล่น	18
3.5 User requirement	18
3.6 Use case diagram	20
3.7 Class diagram	29
3.8 การออกแบบหน้าต่างโปรแกรม	30
บทที่ 4 การทดลองและผลการทดลอง	33
4.1 หน้าเมนูหลัก	33
4.2 หน้าเมนูหลัก	33
4.3 การเคลื่อนที่ของกล้อง	35
4.4 ระบบเงิน	35
4.5 Minimap	36
4.6 แหล่งทรัพยากร	37
4.7 แถบแสดงสถานะ	37
4.8 การหยุดพักเกม	38
4.9 การสร้างสิ่งก่อสร้าง	39
4.10 การสร้างหุ่นยนต์	40
4.11 ศัตรู	42
4.12 การบังคับตัวละคร	42
4.13 การต่อสู้	44
4.14 ภารกิจ	45

สารบัญ (ต่อ)

	หน้า
บทที่ 5 บทสรุปและข้อเสนอแนะ	47
5.1 สรุปผลที่ได้จากการทำโครงการ.....	47
5.2 ปัญหาและอุปสรรค.....	47
5.3 แนวทางในการพัฒนา.....	47
บรรณานุกรม	49



สารบัญตาราง

ตาราง	หน้า
1.1 แผนการดำเนินงาน	2
2.1 ความสัมพันธ์ระหว่าง Prototype stage กับ Quality criteria	6
3.1 ข้อมูลรายละเอียดของคำสั่งการกระทำ	11
3.2 ข้อมูลรายละเอียดของคำสั่งตรวจสอบ	11
3.3 ข้อมูลรายละเอียดของคำสั่งตัวแปร	12
3.4 ข้อมูลรายละเอียดของคำสั่งเงื่อนไขควบคุม	13
3.5 ข้อมูลรายละเอียดของสิ่งก่อสร้าง	14
3.6 ข้อมูลรายละเอียดของหุ่นยนต์	16
3.7 User requirement specification	18
3.8 รายละเอียดของ Use case	21
3.9 รายละเอียดของ Actor	21
3.10 รายละเอียดของ Use Case สร้างบัญชีผู้เล่น	21
3.11 รายละเอียดของ Use Case เลือกรับบัญชีผู้เล่น	22
3.12 รายละเอียดของ Use Case ลบบัญชีผู้เล่น	22
3.13 รายละเอียดของ Use Case เลือกร้าน	23
3.14 รายละเอียดของ Use Case สร้างไฟล์คำสั่ง	23
3.15 รายละเอียดของ Use Case แก้ไขไฟล์คำสั่ง	24
3.16 รายละเอียดของ Use Case ลบไฟล์คำสั่ง	24
3.17 รายละเอียดของ Use Case นำไฟล์คำสั่งเข้าหุ่นยนต์	25
3.18 รายละเอียดของ Use Case การตั้งค่าโรงงาน	25
3.19 รายละเอียดของ Use Case เลือกจำนวนหุ่นยนต์ที่จะสร้าง	26
3.20 รายละเอียดของ Use Case วางสิ่งก่อสร้าง	26
3.21 รายละเอียดของ Use Case ควบคุมความเร็วเกม	27
3.22 รายละเอียดของ Use Case วางและย้ายธง	27
3.23 รายละเอียดของ Use Case ลบธง	28

สารบัญรูป

รูป	หน้า
2.1 เกม StarCraft II	3
2.2 เกม Civilization VI.....	4
2.3 Blockly game คำน maze	5
2.4 โลโก้ Unity 3D.....	6
2.5 Game Development Life Cycle.....	9
3.1 use case diagram	20
3.2 class diagram	29
3.3 หน้าตาโปรแกรมเมื่อคลิกเลือกยูนิท	30
3.4 หน้าตาโปรแกรมเมื่อคลิกเลือกโรงประกอบหุ่นยนต์	31
3.5 หน้าตาโปรแกรมเมื่อสร้างสิ่งก่อสร้าง	31
3.6 หน้าตาโปรแกรมเมื่อเขียนโปรแกรมคำสั่ง	32
4.1 โปรแกรม Unity3D	33
4.2 หน้าต่างเมนูหลัก	34
4.3 หน้าต่าง Option	34
4.4 หน้าต่างเลือกด้านภายในเกม	35
4.5 หน้าต่างแสดงจำนวนเงินที่มีอยู่.....	35
4.6 จำนวนขีดจำกัดในการเก็บเงินเมื่อยังไม่มี Warehouse	36
4.7 จำนวนขีดจำกัดในการเก็บเงินเมื่อมี Warehouse	36
4.8 หน้าต่าง Minimap	36
4.9 หุ่นยนต์ Miner กำลังขุดแร่	37
4.10 แถบสถานะเมื่อเลือกหุ่นยนต์ประเภท Attack	37
4.11 แถบสถานะเมื่อเลือกหุ่นยนต์ประเภท General	38
4.12 แถบสถานะเมื่อเลือกที่สิ่งก่อสร้าง	38
4.13 การหยุดพักเกม	38
4.14 หน้าต่างการสร้างสิ่งก่อสร้าง	39
4.15 การสร้างสิ่งก่อสร้าง	40

สารบัญรูป (ต่อ)

รูป	หน้า
4.16 หน้าต่างการสร้างหุ่นยนต์	41
4.17 หุ่นยนต์ที่สร้างออกมาอยู่ล้อมรอบ Factory.....	41
4.18 หุ่นยนต์และสิ่งก่อสร้างผังศัตรู	42
4.19 การเลือกตัวละคร 1 ตัว	43
4.20 การสั่งให้ตัวละครเคลื่อนที่	43
4.21 การเลือกตัวละครหลายตัว.....	44
4.22 การต่อสู้กันระหว่างหุ่นยนต์ทั้ง 2 ฝ่าย	44
4.23 หุ่นยนต์ฟันผู้เล่นโจมตีสิ่งก่อสร้างศัตรู	45
4.24 หน้าต่างภารกิจ.....	45
4.25 หน้าต่างบรรณภารกิจ เมื่อทำภารกิจสำเร็จหมด.....	46



บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

ในยุคสมัยปัจจุบันมีสื่อต่างๆ ที่คอยให้ความบันเทิงและประโยชน์ต่างๆ อยู่เป็นจำนวนมาก ไม่ว่าจะเป็นรูปของหนังสือ วิดีโอ เพลง และสื่ออีกประเภทหนึ่งคือเกม ซึ่งเป็นสื่อสมัยใหม่ที่ได้รับ ความนิยมและความสนใจจากผู้คนเป็นจำนวนมาก ทำให้เกิดเป็นอาชีพต่างๆ ที่ทำงานเกี่ยวกับสื่อ เกม อาทิเช่น นักกีฬาอีสปอร์ต นักพากย์เกม นักวิเคราะห์เกม หรือนักพัฒนาเกม โดยนักพัฒนาเกม เป็นอาชีพที่มีความสามารถในการสร้าง ออกแบบ และพัฒนา โปรแกรมเกมขึ้นมาได้ โดยใช้องค์ ความรู้ต่างๆ เช่น การเขียน โปรแกรมคอมพิวเตอร์ การออกแบบ โครงสร้างและหน้าตาของ โปรแกรมเกม เป็นต้น และเนื่องจากทางคณะผู้จัดทำได้มีความสนใจในการศึกษาวิธีการพัฒนา โปรแกรมเกมและต้องการพัฒนาโปรแกรมให้ออกมามีคุณภาพและสร้างความเพลิดเพลิน สนุกสนานให้แก่ผู้เล่น จึงเป็นที่มาของ โครงการงานที่ทำการพัฒนาโปรแกรมเกมขึ้นมาให้มีคุณภาพ มี ความน่าสนใจและดึงดูดผู้เล่น อีกทั้งให้ความบันเทิงและกระบวนการคิดวางแผนให้แก่ผู้ใช้งาน

1.2 วัตถุประสงค์ของโครงการงาน

- 1) เพื่อศึกษากระบวนการสร้าง ออกแบบ และพัฒนา โปรแกรมเกมให้ออกมามีคุณภาพและให้ ความเพลิดเพลินแก่ผู้เล่น
- 2) เพื่อพัฒนาโปรแกรมเกมสำหรับให้ความบันเทิงแก่ผู้ใช้งาน
- 3) เพื่อศึกษาและฝึกการใช้งาน โปรแกรม Unity 3D

1.3 ขอบเขตของโครงการงาน

- 1) โปรแกรมเกมสามารถทำงานบนแพลตฟอร์ม ไมโครซอฟท์วินโดวส์
- 2) โปรแกรมเกมสามารถเล่นได้คนเดียว
- 3) โปรแกรมเกมเป็นเกมออฟไลน์ ไม่มีการเชื่อมต่ออินเทอร์เน็ต
- 4) ผู้เล่นสามารถเลือกด่านต่างๆ ของเกมได้เมื่อได้เล่นจบด่านนั้นๆ แล้ว
- 5) ผู้เล่นสามารถสร้างและออกคำสั่งกับสิ่งก่อสร้างและยูนิตต่างๆ ในเกมได้ตามที่กำหนดไว้
- 6) ผู้เล่นสามารถทำภารกิจภายในเกมได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้โปรแกรมเกมวางแผนการรบที่ช่วยให้ผู้เล่นได้รับความสนุกเพลิดเพลิน
- 2) ได้รับความรู้และทักษะในการออกแบบและพัฒนาโปรแกรมเกม
- 3) ได้รับความรู้และทักษะในการใช้งานโปรแกรมและอุปกรณ์เกี่ยวกับการออกแบบและพัฒนาโปรแกรมเกม

1.5 แผนการดำเนินงาน

ตาราง 1.1 แผนการดำเนินงาน

งานที่ทำ	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
กำหนดหัวข้อและขอบเขตของโครงการ									
ศึกษาการใช้งาน Unity 3D และการออกแบบเกม									
ออกแบบโปรแกรมเกม									
ทดลองเขียนโปรแกรมตามที่ได้ออกแบบไว้									
ทำเล่มรายงาน									
พัฒนาโปรแกรมเกม									

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 เกม Strategy

เกม Strategy เป็นประเภทของวิดีโอเกมประเภทหนึ่ง มีจุดเด่นในด้านการใช้ความคิดและการวางแผนต่างๆ เพื่อให้บรรลุเป้าหมายตามที่เกมกำหนดไว้ โดยเฉพาะการที่ผู้เล่นต้องวางแผนการดำเนินการกับฝ่ายตรงข้ามอย่างน้อยหนึ่งฝ่าย ส่วนใหญ่เป็นการลดจำนวนกองกำลังของฝ่ายตรงข้าม ซึ่งจะส่งผลต่อการบรรลุเป้าหมายของเกม มุมมองการเล่นจะเป็นการมองจากด้านบนและเราจะทำการสร้างหรือควบคุมสิ่งก่อสร้างและยูนิตด้วยคำสั่งต่างๆ ภายในเกม โดยแนวทางของเกมจะเป็นเกี่ยวกับสงครามการสู้รบในสถานการณ์จำลองที่แตกต่างกัน และนอกจากการสู้รบแล้วยังมีการสำรวจและจัดการทรัพยากรให้เพียงพอต่อการใช้งาน เราสามารถแบ่งย่อยประเภทของเกม Strategy ได้ออกเป็น 2 ประเภทใหญ่ๆ ได้แก่

2.1.1 Real-time strategy (RTS)

ลักษณะเด่นของแนวเกมนี้คือเกมจะดำเนินอยู่ตลอดเวลา ไม่ได้ดำเนินแบบแบ่งเป็นเทิร์นจะมุ่งเน้นไปที่การวางตำแหน่งของสิ่งก่อสร้างและการสร้างยูนิตออกมาเพื่อใช้ในการป้องกันอาณาเขตฝั่งเราหรือเพื่อทำลายหรือยึดอาณาเขตของฝั่งตรงข้าม โดยแตกต่างกับการเสียทรัพยากรตามที่กำหนดไว้ ทำให้ต้องมีการควบคุมจัดการให้ทรัพยากรเพียงพอแก่การใช้งาน ตัวอย่างของวิดีโอเกมที่เป็นรูปแบบ RTS เช่น StarCraft II Warhammer 40,000:Dawn of War III Age of Empires III เป็นต้น



รูป 2.1 เกม StarCraft II

2.1.2 Turn-based strategy (TBS)

รูปแบบการเล่นของแนวเกมจะใกล้เคียงกับแบบ Real-time strategy จะแตกต่างที่จะทำการแบ่งเวลาการเล่นของผู้เล่นออกเป็นช่วงๆ เรียกว่า เทิร์น โดยในระหว่างเทิร์นนั้น เกมจะทำการหยุดเพื่อให้ผู้เล่นได้คิดวิเคราะห์วางแผนและดำเนินการตามคำสั่งที่สามารถทำได้ในเทิร์นนั้น เมื่อหมดเทิร์นของเราจะสลับไปเป็นเทิร์นของอีกฝั่งไปเรื่อยๆ จนกว่าจะมีฝ่ายใดบรรลุเป้าหมายของเกมได้ ตัวอย่างวิดีโอเกมที่เป็นรูปแบบ TBS เช่น Civilization VI XCOM 2 เป็นต้น



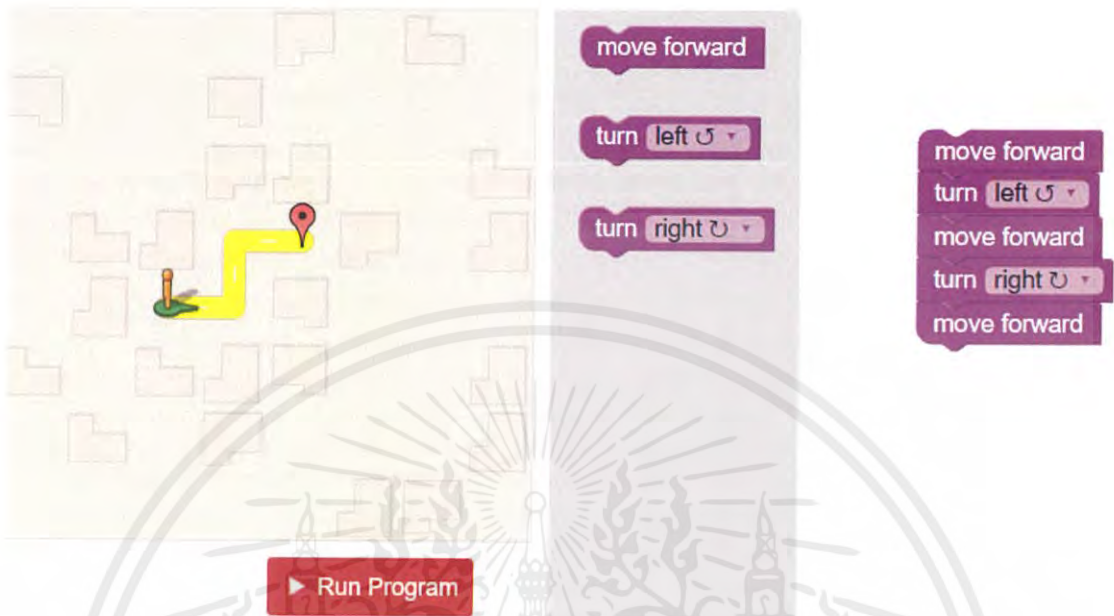
รูป 2.2 เกม Civilization VI

เกมประเภท Strategy มีประโยชน์ในด้านการพัฒนาสมองให้มีความคล่องตัวและปรับปรุงการคิดเชิงกลยุทธ์ของผู้เล่นให้มีประสิทธิภาพมากขึ้น อ้างอิงจากงานวิจัยของ Queen Mary University of London และ University College London โดยแบ่งผู้ร่วมการทดลองออกเป็นกลุ่มที่ให้เล่นเกม StarCraft II ซึ่งเป็นเกมประเภท Real-time Strategy และกลุ่มที่ให้เล่นเกม The Sims ซึ่งเป็นเกมประเภท Simulation เป็นระยะเวลา 40 ชั่วโมงภายใน 6 – 8 สัปดาห์ ผลที่ได้คือกลุ่มที่เล่นเกม StarCraft II จะมีความยืดหยุ่นทางความคิด (cognitive flexibility) ซึ่งเป็นทักษะสมองที่ปรับเปลี่ยนความคิดเปลี่ยนจุดสนใจไปตามที่ต้องการหรือตามสถานการณ์ที่เปลี่ยนไป มากกว่าอีกกลุ่มที่เล่นเกม The Sims

2.2 Blockly Game

blockly game เป็นเกมที่ใช้สำหรับสอนการเขียนโปรแกรมเบื้องต้น พัฒนาโดย Google จุดเด่นคือเกมถูกออกแบบมาสำหรับเด็กหรือผู้ที่ไม่มีประสบการณ์ในการเขียนโปรแกรมเพื่อให้สามารถเรียนรู้และเข้าใจได้โดยง่าย โดยใช้ภาษา blockly มีลักษณะเป็นบล็อกคำสั่ง สามารถนำมาต่อกันเป็น

บรรทัดเพื่อสร้างเป็นโปรแกรมๆหนึ่งได้ ซึ่งเกมนี้เป็นแรงบันดาลใจของทางผู้จัดทำในการสร้าง บล็อกคำสั่งต่างๆในการเขียนโปรแกรมบังคับยูนิตต่างๆ ภายในเกม



รูป 2.3 Blockly game ด้าน maze

2.3 Unity 3D

Unity 3D เป็น cross-platform game engine ที่สร้างขึ้นจากบริษัท Unity Technologies ซึ่งเกมเอนจินนี้สามารถทำการพัฒนาเกมได้ทั้งในรูปแบบสองมิติและสามมิติและสามารถนำโปรแกรมเกมไปลงในแพลตฟอร์มต่างๆ ได้มากมาย ภาษาที่ใช้ในการเขียนสคริปของตัวเกมเป็นหลักจะเป็นภาษา C# จุดเด่นของ Unity คือเป็นเกมเอนจินที่ปล่อยให้ใช้งานฟรี แค่อันดับสามารถพัฒนาเกมที่มีคุณภาพสูงออกมาได้ สามารถพัฒนาเกมได้หลายประเภท อาทิเช่น เกมแอคชั่น เกม FPS เกมปริศนา เกมวางแผนกลยุทธ์ และอีกมากมาย รวมถึงสามารถนำโปรแกรมเกมไปพอร์ตลงได้หลากหลายช่องทาง ไม่ว่าจะเป็นในคอมพิวเตอร์ทั้งระบบปฏิบัติการ Windows หรือ Mac ในเครื่องเล่นเกมคอนโซลในตระกูล PlayStation หรือ Xbox ในโทรศัพท์มือถือระบบปฏิบัติการ iOS และ Android หรือแม้ในเครื่องเล่นเกมพกพาอย่าง Nintendo Switch และอีกมากมาย



รูป 2.4 โลโก้ Unity 3D

2.4 Fullerton's Game Prototype Usability Quality Criteria

เป็นเกณฑ์ที่ใช้ประเมินคุณภาพของ prototype เกมที่พัฒนาขึ้นมา ซึ่งมีเกณฑ์อยู่ทั้งหมด 5 หัวข้อ ได้แก่

2.3.1 Functional

Feature ต่างๆ ภายในเกมต้องสามารถทำงานได้และเป็นทำงานได้ตามที่ได้กำหนดไว้ สามารถทดสอบได้จากความสมบูรณ์ของการทำงานในแต่ละ feature

2.3.2 Internally complete

กฎ กติกา และเงื่อนไขภายในเกมต้องได้รับการจัดการและมีความถูกต้อง สามารถทดสอบได้จากการตรวจสอบและสังเกตข้อบกพร่องระหว่างการเล่นทดสอบ

2.3.3 Balanced

ระดับความยากของเกมมีความเหมาะสม ไม่ยากหรือง่ายเกินไป สามารถทดสอบได้จากการอภิปรายหรือแบบสอบถามเกี่ยวกับระดับความยากของเกม

2.3.4 Fun

เกมต้องทำให้เกิดความสนใจ ความเพลิดเพลิน มีความท้าทาย และสามารถดึงดูดผู้เล่นได้ สามารถทดสอบได้จากการแบบสอบถามหรือสอบถามจากผู้เล่นโดยตรง

2.3.5 Accessible

ผู้เล่นสามารถเรียนรู้และทำความเข้าใจรวมทั้งใช้งานระบบต่างๆ ในเกมได้ง่ายและรวดเร็ว สามารถทดสอบได้จากการสังเกตความสามารถในการเล่นและการควบคุมเกมของผู้เล่น รวมถึงเวลาที่ผู้เล่นใช้ในการเรียนรู้ user interface

ในส่วนของ prototype เกมจะแบ่งออกเป็น 4 ระดับ ได้แก่ Foundation Structure Format details และ Refinement ซึ่งจะอธิบายรายละเอียดในหัวข้อถัดไป และในแต่ละ prototype จะมีเกณฑ์ที่ใช้ในการประเมินแตกต่างกันออกไปตามตารางที่

ตาราง 2.1 ความสัมพันธ์ระหว่าง Prototype stage กับ Quality criteria

Prototype Stage	Functional	Internally complete	Balanced	Fun	Accessible
Foundation				✓	
Structure	✓			✓	
Format details	✓	✓	✓		
Refinement				✓	✓

2.5 Game Development Life Cycle

โปรแกรมเกมถือว่าเป็นซอฟต์แวร์ประเภทหนึ่ง โดยในการพัฒนาซอฟต์แวร์ตามปกติจะมีขั้นตอนในการพัฒนาตามโมเดล Software Development Life Cycle ซึ่งเป็นกระบวนการทางด้านวิศวกรรมที่ใช้ในการพัฒนา แต่เนื่องจากโปรแกรมเกมไม่ได้เป็นผลิตภัณฑ์ทางวิศวกรรมเพียงอย่างเดียว แต่มีผสมผสานกันระหว่างศิลปะ ดนตรี การเขียนโปรแกรม การแสดง และการจัดการ ทำให้การใช้ Software Development Life Cycle ในการพัฒนาโปรแกรมเกมทำให้ผู้พัฒนาพบกับปัญหาต่างๆ ทำให้เกิดการพัฒนาลักษณะ Game Development Life Cycle ขึ้นมาใช้ในการพัฒนาโปรแกรมเกมแทน โดยมีขั้นตอนดังต่อไปนี้

2.5.1 Initiation

เป็นขั้นตอนในการคิดว่าจะเลือกทำเกมอะไร เป็นเกมแนวไหน มี game concept คร่าวๆ เป็นอย่างไร โดยผลลัพธ์ที่ได้เป็น game concept และรายละเอียดเกมแบบคร่าวๆ

2.5.2 Pre-production

เป็นขั้นตอนที่มีความสำคัญ ในขั้นตอนนี้จะเป็นการสร้างและออกแบบ game design ซึ่งจะมุ่งเน้นในการกำหนดประเภทของเกม เกมการเล่น กลไก เนื้อเรื่อง ตัวละคร ความท้าทาย บังคับที่ทำให้เกมสนุกสนาน องค์ประกอบทางด้านเทคนิค และองค์ประกอบของเอกสารในเอกสารการออกแบบเกม หลังจากนั้นจะเป็นการสร้าง game prototype ตาม game design ที่ได้ทำมา โดยแบ่ง prototype ออกเป็น 2 ช่วงคือ

2.5.2.1 Foundation

เป็น prototype ขั้นแรก จะเป็นการแสดงตัวอย่างเกมว่ามีเกมการเล่นหลักเป็นอย่างไรและสามารถทำอะไรได้บ้างในเกม

2.5.2.2 Structure

เป็น prototype ขั้นต่อไปที่มีรายละเอียดเพิ่มเติมจาก Foundation โดยเพิ่มเติมในส่วนเกมการเล่นหลักและส่วนกลไกการเล่นต่างๆ ที่เกี่ยวข้องการคำนวณ ตรรกศาสตร์ และกฎกติกาของเกม

2.5.3 Production

เป็นขั้นตอนที่มีความสำคัญต่อของวงจร จะเป็นการสร้างเนื้อหา ซอร์สโค้ด และองค์ประกอบที่เกี่ยวข้องของทั้งสองอย่าง โดย prototype จะได้รับการพัฒนาอีก 2 ขั้นตอนได้แก่

2.5.3.1 Formal Details

เป็น prototype ที่มีรายละเอียดมากกว่า Structure โดยเพิ่มกลไกเกมแบบเสร็จสมบูรณ์และเนื้อหาต่างๆ มีการปรับสมดุลเกมให้เหมาะสม เพิ่ม feature ใหม่ ปรับปรุงประสิทธิภาพโดยรวม และแก้ไขบั๊กของโปรแกรมเกม

2.5.3.2 Refinement

เป็น prototype ที่เสร็จสมบูรณ์แล้ว จะมีเพิ่มรายละเอียดของเกมโดยทำให้เกมมีความสนุกมากขึ้น ทำทายมากขึ้นและเข้าใจได้ง่าย ในขั้นตอนนี้จะสามารถเปลี่ยนแปลงตัวเกมได้เพียงเล็กน้อยเท่านั้น

2.5.4 Testing

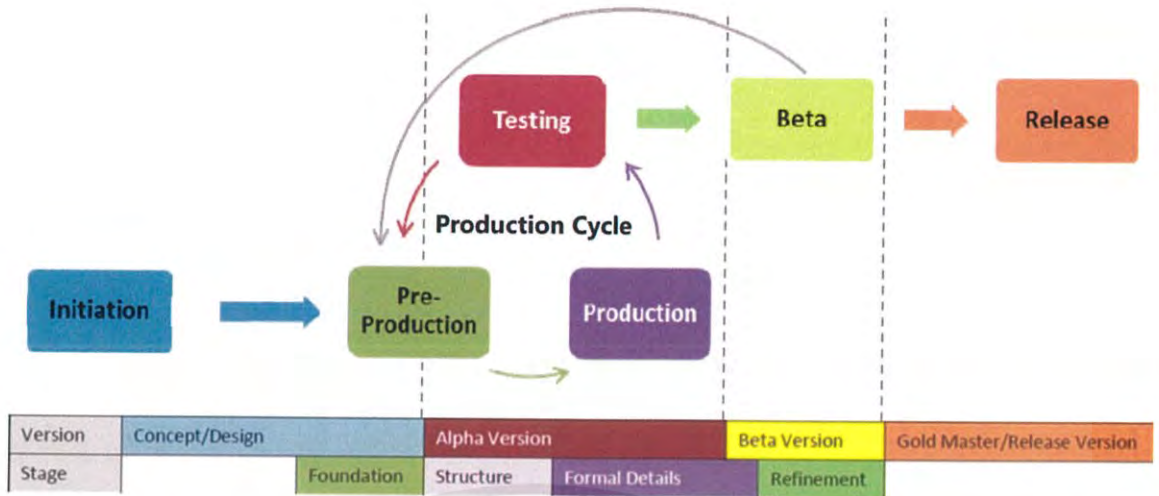
เป็นการทดสอบที่ให้ tester ในทีมพัฒนาเป็นคนทดสอบ โดยมีการทดสอบความสามารถในการใช้งานและการเล่น แบ่งย่อยออกเป็น Formal Detail testing เป็นการทดสอบในส่วนของการใช้งานและระดับความยากของเกม และ Refinement Testing เป็นการทดสอบในส่วนของคุณภาพของเกมและความเข้าถึงง่ายของตัวเกม ผลลัพธ์ที่ได้จากขั้นตอนนี้คือรายงานการเกิดบั๊ก คำขอการเปลี่ยนแปลง และการตัดสินใจในการพัฒนา ซึ่งจะมีผลต่อการตัดสินใจว่าจะไปถึงขั้นตอนต่อไป (Beta) หรือจะอยู่ในขั้นตอนวงจร Production ต่อไป

2.5.5 Beta

เป็นขั้นตอนที่ให้ tester ที่เป็นคนนอกทีมพัฒนาเป็นผู้ทดสอบ โดยยังใช้หลักการทดสอบเหมือนเดิมกับขั้นตอนก่อนหน้า จะมีการเลือก tester แบ่งเป็น 2 ประเภทคือ closed beta ซึ่งจะเฉพาะบุคคลที่ได้รับเชิญเท่านั้น และ open beta ซึ่งบุคคลทั่วไปสามารถเข้าร่วมการทดสอบได้

2.5.6 Release

เป็นขั้นตอนสุดท้ายที่จะทำการปล่อยตัวเกมออกสู่สาธารณะพร้อมทั้งเอกสาร ข้อมูลต่างๆ สรุปผลโครงการ และแผนการในการปรับปรุงแก้ไขและระบบใหม่ๆ เพิ่มเติมในอนาคต



รูป 2.5 Game Development Life Cycle



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการพัฒนา

3.1 ระบบการเล่นหลัก

3.1.1 ภาพรวม

จุดประสงค์ของเกมนี้คือการก่อตั้งฐานสำรวจ/ฐานทัพเพื่อทำภารกิจ โดยการวางระบบ หุ่นยนต์อัตโนมัติที่สามารถเขียนชุดคำสั่งให้ทำงานตามหน้าที่แต่ละส่วนของฐาน หรือบรรลุมิติวัตถุประสงค์ในแต่ละส่วนของภารกิจ รวมถึงต้องเอาชนะอุปสรรคต่างๆภายในเกม

ระบบเวลาในเกมจะดำเนินเป็นคาบเวลา หรือ Cycle ซึ่งในแต่ละ Cycle หุ่นยนต์แต่ละตัว ทั้งฝ่ายผู้เล่นและฝ่ายศัตรู ทำคำสั่งของตนเองจนถึงคำสั่งการกระทำ (Action) ซึ่งเป็นคำสั่งที่หุ่นยนต์ ปฏิสัมพันธ์กัน ทั้งนี้หุ่นยนต์ทุกตัวจะประมวลผลและทำงานพร้อมกัน ผู้เล่นสามารถคุมความเร็วในการดำเนินเกมได้ โดยจะให้เร็วขึ้น ช้าลง หยุดชั่วคราวหรือให้ดำเนินไปที่ละ Cycle เพื่อให้ผู้เล่นมีโอกาสจัดการบริหาร รวมถึงให้ผู้เล่นทำการดีบัคคำสั่งของหุ่นยนต์ได้

หุ่นยนต์จะมีความสามารถ ความเร็ว ความทนทาน (HP) และเกราะแตกต่างกันตาม ประเภทของหุ่นยนต์ มีหน่วยความจำสามารถเก็บข้อมูลเดี่ยวได้ 16 ตัวและเก็บข้อมูลเป็นลิสต์ได้ 4 ลิสต์ ซึ่งเก็บข้อมูลได้ประเภทเดียวต่อลิสต์ โดยหุ่นยนต์จะจำผลลัพธ์เอาที่พูดครั้งล่าสุดลง หน่วยความจำชั่วคราวเสมอ มีขีดจำกัดจำนวนคำสั่งที่รันได้ต่อ Cycle ถ้าหุ่นยนต์รันเกินข้อจำกัด จะถือว่าหุ่นตัวนั้นไม่พร้อมทำ Action ใน Cycle นั้น

3.1.2 ระบบการเขียนคำสั่ง

ระบบการเขียนคำสั่งเป็นหนึ่งในระบบการเล่นหลักของเกม โดยที่ผู้เล่นเขียนชุดคำสั่งให้ หุ่นยนต์แต่ละตัวหรือแต่ละกลุ่มบรรลุมิติหรือจุดประสงค์ย่อยๆ อาทิ เก็บเกี่ยวทรัพยากร ก่อสร้าง ตึก, ป้องกัน/โจมตี เป็นต้น ซึ่งหุ่นยนต์แต่ละตัวจะมีความสามารถต่างกันไปตามประเภทหุ่น แต่มี ข้อจำกัดที่ว่าผู้เล่นไม่สามารถสั่งการให้หุ่นยนต์ทำงานได้โดยตรง

การเขียนคำสั่งจะอยู่ในรูปแบบของการลากและวางบล็อกคำสั่ง เรียบเรียงลงเป็นชุดคำสั่ง ผู้เล่นสามารถเก็บบันทึกชุดคำสั่งเพื่อนำไปใช้งานกับหุ่นยนต์ตัวอื่น หรือนำไปใช้ในภารกิจอื่นๆได้ ซึ่งบล็อกคำสั่งประกอบไปด้วย

3.1.2.1 คำสั่งการกระทำ (Actions)

คำสั่งในกลุ่มนี้เป็นกลุ่มเดียวที่จะใช้เวลาเป็น Cycle ในการทำคำสั่งและได้ผลลัพธ์ออกมา คือความเปลี่ยนแปลงของตนเอง หุ่นยนต์ตัวอื่น สภาพแวดล้อม หรือสถานการณ์ภายในเกม คำสั่งในกลุ่มการกระทำมีดังนี้

ตาราง 3.1 ข้อมูลรายละเอียดของคำสั่งการกระทำ

ชื่อ	รายละเอียด
MoveTo	เดินไปยังตำแหน่งที่ระบุไว้ด้วยระยะทางที่กำหนด
MoveBack	ถอยห่างจากตำแหน่งที่ระบุไว้ด้วยระยะทางที่กำหนด
Mine	ทำการขุดแร่จากเป้าหมาย โดยตัวหุ่นจะหยิบแร่ขึ้นโดยอัตโนมัติ
PickUp	หยิบไอเท็มเป้าหมาย สามารถสั่งให้นำไอเท็มออกจากตึกได้
DropAt	วางไอเท็มที่ถือไว้ สามารถสั่งให้ไอเท็มเก็บเข้าตึกได้
Build	ทำการก่อสร้างตึกเป้าหมาย
Repair	ทำการซ่อมบำรุงตึกเป้าหมาย
Attack	สั่งโจมตีเป้าหมาย
Wait	สั่งให้หุ่นรอ ไม่ทำการใดๆ

3.1.2.2 คำสั่งตรวจสอบ (Checks)

คำสั่งในกลุ่มนี้ตรวจสอบสภาพแวดล้อม หรือสถานต่างของวัตถุภายในเกม แล้วจะได้ผลลัพธ์ออกมาเป็นข้อมูลต่างๆที่จะเก็บไว้ในหน่วยความจำชั่วคราวของหุ่น มีดังนี้

ตาราง 3.2 ข้อมูลรายละเอียดของคำสั่งตรวจสอบ

ชื่อ	รายละเอียด
ScanAll	ตรวจหาวัตถุตามแท็กที่กำหนดไว้ภายในระยะที่กำหนด เอาท์พุทเป็นลิสต์ของวัตถุ
ScanNearest	ตรวจหาวัตถุตามแท็กที่กำหนดไว้ ภายในระยะที่กำหนด เอาท์พุทเป็นวัตถุที่อยู่ใกล้สุด
ScanFarthest	ตรวจหาวัตถุตามแท็กที่กำหนดไว้ ภายในระยะที่กำหนด เอาท์พุทเป็นวัตถุที่อยู่ไกลสุดที่
GetHPPercent	ตรวจสอบ HP ของวัตถุ เอาท์พุทเป็นตัวเลขเปอร์เซ็นต์ จาก 1 ถึง 100 (เต็ม)

ตาราง 3.2 ข้อมูลรายละเอียดของคำสั่งตรวจสอบ (ต่อ)

GetPosition	ตรวจสอบตำแหน่งของวัตถุ เอาท์พุทเป็นพิกัดจุด
CheckTag	ตรวจสอบแท็กของวัตถุเทียบกับเงื่อนไข เอาท์พุทเป็นบูลีนจริงถ้ามีแท็กตรงตามที่ตรวจสอบทั้งหมด
CheckFlagPlaced	ตรวจสอบสถานะของธงที่มีสัญลักษณ์และเบอร์ที่กำหนด เอาท์พุทเป็นบูลีนจริงถ้าที่ธงที่กำหนดปักอยู่
GetFlagPosition	ตรวจสอบตำแหน่งของธงที่มีสัญลักษณ์และเบอร์ที่กำหนด เอาท์พุทเป็นพิกัดจุด
GetFlagList	เอาท์พุทลิสต์ของธงที่มีสัญลักษณ์ตามที่กำหนด

3.1.2.3 คำสั่งตัวแปร (Variables)

คำสั่งในกลุ่มนี้จัดการข้อมูลและหน่วยความจำต่างๆภายในตัวหุ่นยนต์ รวมถึงการอ้างอิงต่างๆ มีดังนี้

ตาราง 3.3 ข้อมูลรายละเอียดของคำสั่งตัวแปร

ชื่อ	รายละเอียด
StoreToMem	นำค่าเอาท์พุทล่าสุดเก็บไว้ในตัวแปรเดี่ยวที่กำหนด
ReadMem	อ้างอิงค่าที่เก็บไว้ในตัวแปรเดี่ยวที่กำหนด
AddToList	นำค่าเอาท์พุทล่าสุดเก็บเข้าในลิสต์ที่กำหนด
RemoveFromList	ลบค่าที่ระบุไว้ในลิสต์ออก
ListCount	เอาท์พุทจำนวนข้อมูลจากลิสต์ที่กำหนด
FirstInList	อ้างอิงค่าที่เก็บไว้ในตัวแรกสุดในลิสต์ที่กำหนด
LastInList	อ้างอิงค่าที่เก็บไว้ในตัวท้ายสุดในลิสต์ที่กำหนด
nInList	อ้างอิงค่าที่เก็บไว้ในลำดับที่ระบุในลิสต์ที่กำหนด
ClearList	ลบข้อมูลลิสต์ที่ระบุออกทั้งหมด

3.1.2.4 เงื่อนไขควบคุม (Flow Control)

คำสั่งในกลุ่มนี้ปรับเปลี่ยนการทำงานของหุ่นยนต์ขึ้นอยู่กับเงื่อนไขที่ให้

ตรวจสอบ

ตาราง 3.4 ข้อมูลรายละเอียดของคำสั่งเงื่อนไขควบคุม

ชื่อ	รายละเอียด
If-Then / If-Then-Else	ตรวจสอบเงื่อนไขว่าเป็นจริงไหม แล้วทำคำสั่งสอดคล้องตามผลลัพธ์ที่ได้
Repeat-n-Time	วนทำคำสั่งซ้ำตามจำนวนครั้งที่ระบุ
While loop	วนทำคำสั่งซ้ำตามเมื่อเงื่อนไขเป็นจริง

3.1.3 ระบบธง (Flags)

ธงในเกมทำหน้าที่เป็นจุดอ้างอิงให้หุ่นยนต์ซึ่งผู้เล่นสามารถปรับตำแหน่งในเกม เคลื่อนย้าย หรือถอนออกได้ โดยหุ่นยนต์สามารถตรวจสอบสถานะของธง และตำแหน่งของธงได้ และนำไปประกอบกับเงื่อนไขควบคุม เพื่อตัดสินใจการกระทำของตัวหุ่นยนต์ตามคำสั่งที่เขียนลง หุ่นยนต์ ผู้เล่นสามารถเลือกสัญลักษณ์และเบอร์ธงที่จะปักในเกมได้ ซึ่งขึ้นอยู่กับว่าหุ่นยนต์มีคำสั่งตอบสนองอย่างไรต่อธงนั้นๆ

3.1.4 ระบบแท็ก (Tags)

หุ่นยนต์ ตึก ไอเท็ม และวัตถุอื่นๆในเกมจะมีลิสต์ของแท็กที่จะระบุตัวตนและสถานะของตัววัตถุ อาทิเช่น เป็นวัตถุอะไร อยู่ฝ่ายไหน มีความเสียหายไหม เป็นต้น ซึ่งสามารถนำไปใช้กรองผลการสแกนของหุ่นยนต์และการประมวลผลอื่นๆได้

3.1.5 การก่อสร้างตึก

กระบวนการก่อสร้างตึกเริ่มต้นจากผู้เล่นวางแปลนสิ่งก่อสร้างเพื่อตั้งเขตก่อสร้างให้ หุ่นยนต์มาทำงาน โดยหุ่นยนต์ต้องขนแร่จากคลังแร่มายังเขตก่อสร้างและทำงานจนครบตามความต้องการในการสร้างตึกนั้นๆให้เสร็จสมบูรณ์

3.1.6 การประกอบหุ่นยนต์

การประกอบหุ่นยนต์จะต้องมีโรงประกอบหุ่นยนต์ที่สร้างไว้แล้วตามขั้นตอนข้างต้น โดยขั้นแรกทำการตั้งค่าโรงประกอบหุ่นยนต์ว่าจะให้ประกอบหุ่นประเภทอะไร และทำการอัพโหลดไฟล์คำสั่งอะไรลงหุ่นยนต์ที่ประกอบเสร็จใหม่ โดยหนึ่งโรงสามารถตั้งค่าการประกอบได้แบบเดียวแต่ปรับเปลี่ยนได้ภายหลังไม่จำกัดครั้ง ขั้นที่สอง ผู้เล่นสามารถตั้งโควต้าว่าจะให้โรงประกอบหุ่นยนต์นั้นสร้างหุ่นยนต์เป็นจำนวนเท่าไร โดยการประกอบจริงจะเริ่มขึ้นก็ต่อเมื่อมีแร่ (ซึ่งขนส่งโดยหุ่นยนต์) มารวบรวมที่โรงประกอบเพียงพอในการสร้างหุ่นยนต์หนึ่งตัวและมีโควต้า กำหนดไว้ที่หนึ่งตัวขึ้นไป

3.2 ข้อมูลรายละเอียดของไอเทมภายในเกม

3.2.1 สิ่งก่อสร้าง

ตาราง 3.5 ข้อมูลรายละเอียดของสิ่งก่อสร้าง

รูป	ชื่อ	รายละเอียด
	ยานคุมภารกิจ Lander MC	<ul style="list-style-type: none"> - ปลดล็อกและสามารถสร้าง WorkerBot (หุ่นแรงงาน) ได้ - เป็นจุดยุทธศาสตร์ที่ต้องคอยป้องกันไว้ (ต้องมีอยู่อย่างน้อย 1 ตึก) - ทำหน้าที่เป็นคลังแร่ หุ่นยนต์สามารถนำแร่เก็บและนำออกไปใช้ได้ *สิ่งก่อสร้างขั้นสูง
	โรงประกอบ หุ่นยนต์ Factory (5 Gold)	<ul style="list-style-type: none"> - สามารถตั้งประกอบหุ่นยนต์ที่ปลดล็อกแล้วได้ - ต้องมีหุ่นยนต์ขั้นแรมารวบรวมที่โรงงานเพื่อทำการประกอบหุ่นยนต์
	โรงก่อสร้าง Builder Bay (10 Gold)	<ul style="list-style-type: none"> - ปลดล็อกหุ่นโยธา - ปลดล็อกสิ่งก่อสร้างขั้นสูง - สร้างได้ 1 หลัง
	โรงแปรรูปแร่ Refinery (20 Gold)	<ul style="list-style-type: none"> - สามารถแปรรูปแร่ดิบเพื่อเพิ่มมูลค่าได้ - ปลดล็อกหุ่นขุดแร่ - สร้างได้ 1 หลัง *สิ่งก่อสร้างขั้นสูง

ตาราง 3.5 ข้อมูลรายละเอียดของสิ่งก่อสร้าง (ต่อ)

	โรงเก็บแร่ Warehouse (25 Gold)	<ul style="list-style-type: none"> - เพิ่มขีดจำกัดของคลังแร่ - ทำหน้าที่เป็นคลังแร่ หุ่นยนต์สามารถนำแร่เก็บและนำออกไปใช้ได้
	คลังอาวุธ Armory (20 Gold)	<ul style="list-style-type: none"> - ปลดล็อกหุ่นยนต์สู้รบ - สร้างได้ 1 หลัง *สิ่งก่อสร้างชั้นสูง
	แลปภาคสนาม Frontier Lab (20 Gold)	<ul style="list-style-type: none"> - ปลดล็อกหุ่นยนต์สนับสนุน - สร้างได้ 1 หลัง *สิ่งก่อสร้างชั้นสูง

3.2.2 หุ่นยนต์

ตาราง 3.6 ข้อมูลรายละเอียดของหุ่นยนต์

รูป	ชื่อ	รายละเอียด
	หุ่นแรงงาน WorkerBot	หุ่นพื้นฐานสร้างง่าย มีความสามารถครอบคลุม แต่ทำงานด้อยกว่าหุ่นยนต์ที่ทำงานเฉพาะด้าน: - ความทนทานต่ำแต่เคลื่อนที่เร็ว - ก่อสร้างได้ - ชุคแร่ได้ - ขนของและขนแร่ได้
	หุ่นโยธา BuilderBot	ออกแบบมาเพื่อก่อสร้างและซ่อมบำรุงโดยเฉพาะ - ความทนทานต่ำ เคลื่อนที่เร็วปานกลาง - ก่อสร้างได้ดี - ขนแร่ได้ - ซ่อมบำรุงดี - รีดอนดีได้ * ต้องมีโรงก่อสร้างก่อน
	หุ่นชุคแร่ MinerBot	ออกแบบมาเพื่อชุคแร่และขนส่งแร่ไปแปรรูป - ความทนทานปานกลาง เคลื่อนที่ช้า - ชุคแร่ได้ดี - ขนแร่ได้ในปริมาณมาก * ต้องมีโรงแปรรูปแร่ก่อน
	หุ่นจู่โจมประชิด MeleeBot	ออกแบบมาเป็นแนวหน้าของกองรบ - มีเกราะและความทนทานสูง - มีอาวุธโจมตีระยะประชิด * ต้องมีคลังอาวุธก่อน

ตาราง 3.6 ข้อมูลรายละเอียดของหุ่นยนต์ (ต่อ)

	หุ่นรถถัง TankBot	ออกแบบมาเป็นพลกำลังหลัก - มีเกราะและความทนทานปานกลาง - มีอาวุธ โจมตีระยะกลาง * ต้องมีคลังอาวุธก่อน
	หุ่นปืนใหญ่ ระยะไกล MortarBot	ออกแบบมาเป็นแนวหลังของกองรบ - มีเกราะและความทนทานต่ำ - มีอาวุธ โจมตีระยะไกล ทำความเสียหายสูงแต่ใช้ เวลายิงนานกว่าหุ่นยนต์รบอื่นๆ * ต้องมีคลังอาวุธก่อน
	หุ่นซ่อมภาคสนาม RepairBot	หุ่นสนับสนุนกองรบ - มีความทนทานปานกลาง แต่ไม่มีเกราะ - ซ่อมบำรุงหุ่นยนต์อื่นๆ ได้ * ต้องมีແປภาคสนามก่อน
	หุ่นปืนกาว GlueBot	หุ่นสนับสนุนกองรบ - มีความทนทานปานกลาง และมีเกราะเบา - สามารถยิงศัตรูเพื่อชะลอความเร็วได้ * ต้องมีແປภาคสนามก่อน

3.3 เนื้อเรื่องเกม

ผู้เล่นจะได้รับบทบาทเป็นนักสำรวจอวกาศที่ทำหน้าที่ควบคุมยานอวกาศจากระยะไกลอยู่ที่โลก เป้าหมายคือไปสำรวจดาวดวงหนึ่งและนำทรัพยากรบนดาวดวงนั้นกลับมา แต่เนื่องจากสภาพอากาศบนดาวเป็นอันตรายต่อมนุษย์ ทำให้ต้องควบคุมยานอวกาศและหุ่นยนต์เก็บทรัพยากรจากระยะไกล

ในระหว่างที่เราเก็บทรัพยากรอยู่ ได้มีกองทัพจักรกลมาบุกทำลายหุ่นยนต์และแย่งชิงทรัพยากรของเรา เราต้องสั่งให้ยานอวกาศหนีออกจากดาวดวงนั้น และเตรียมพร้อมในการกลับไปสู้รบกับกองทัพจักรกลเพื่อแย่งชิงทรัพยากรของดาวดวงนั้น

3.4 เกมการเล่น

ภายในเกมจะมีจำนวนด่านทั้งหมด 5 ด่าน โดยแต่ละด่านจะมีภารกิจหลักที่แตกต่างกันตามด่าน อาทิเช่น เก็บทรัพยากรหรือวัตถุให้ได้ตามที่กำหนด ทำลายสิ่งก่อสร้างที่เป็นแกนหลักของศัตรู เมื่อสามารถทำภารกิจหลักเสร็จสิ้นจะเป็นการจบด่านนั้นไป

สำหรับความยากของเกมจะขึ้นกับด่านที่เล่น โดยด่านในช่วงแรก จะเป็นการสอนพื้นฐานการเล่นต่างๆควบคู่ไปกับเกมการเล่นด้วย อาทิเช่น สอนสร้างสิ่งก่อสร้าง สอนการสั่งคำสั่งให้ยูนิต สอนปุ่มต่างๆในหน้า user interface และจะมียูนิตศัตรูที่จะบุกมาทำลายฐานทัพของเราจำนวนน้อย และมีประเภทของศัตรูไม่หลากหลาย รวมถึงชนิดของสิ่งก่อสร้างและยูนิตของเราจะยังมีจำนวนไม่มาก แต่เมื่อถึงด่านในช่วงกลางถึงท้าย จะเริ่มการเพิ่มจำนวนและชนิดของยูนิตศัตรูมากขึ้น เช่นเดียวกับฝั่งเราที่จะมียูนิตและสิ่งก่อสร้างหลากหลายประเภทขึ้น เพื่อใช้ในการต่อสู้และป้องกัน ฝั่งศัตรูได้มีประสิทธิภาพมากขึ้น

เงื่อนไขในการชนะด่านคือ ทำภารกิจหลักของด่านให้ได้ เช่น ทำลายฐานทัพศัตรู เก็บวัตถุตามสถานที่ต่างๆให้ได้ตามที่กำหนด ทำลายสิ่งก่อสร้างที่กำหนดให้ได้ภายในเวลาที่กำหนด เป็นต้น และเราต้องป้องกันฐานทัพของเราไม่ให้สิ่งก่อสร้างที่เป็นแกนกลางของฐานทัพถูกทำลายก่อนที่จะทำภารกิจหลักเสร็จ

3.5 User requirement

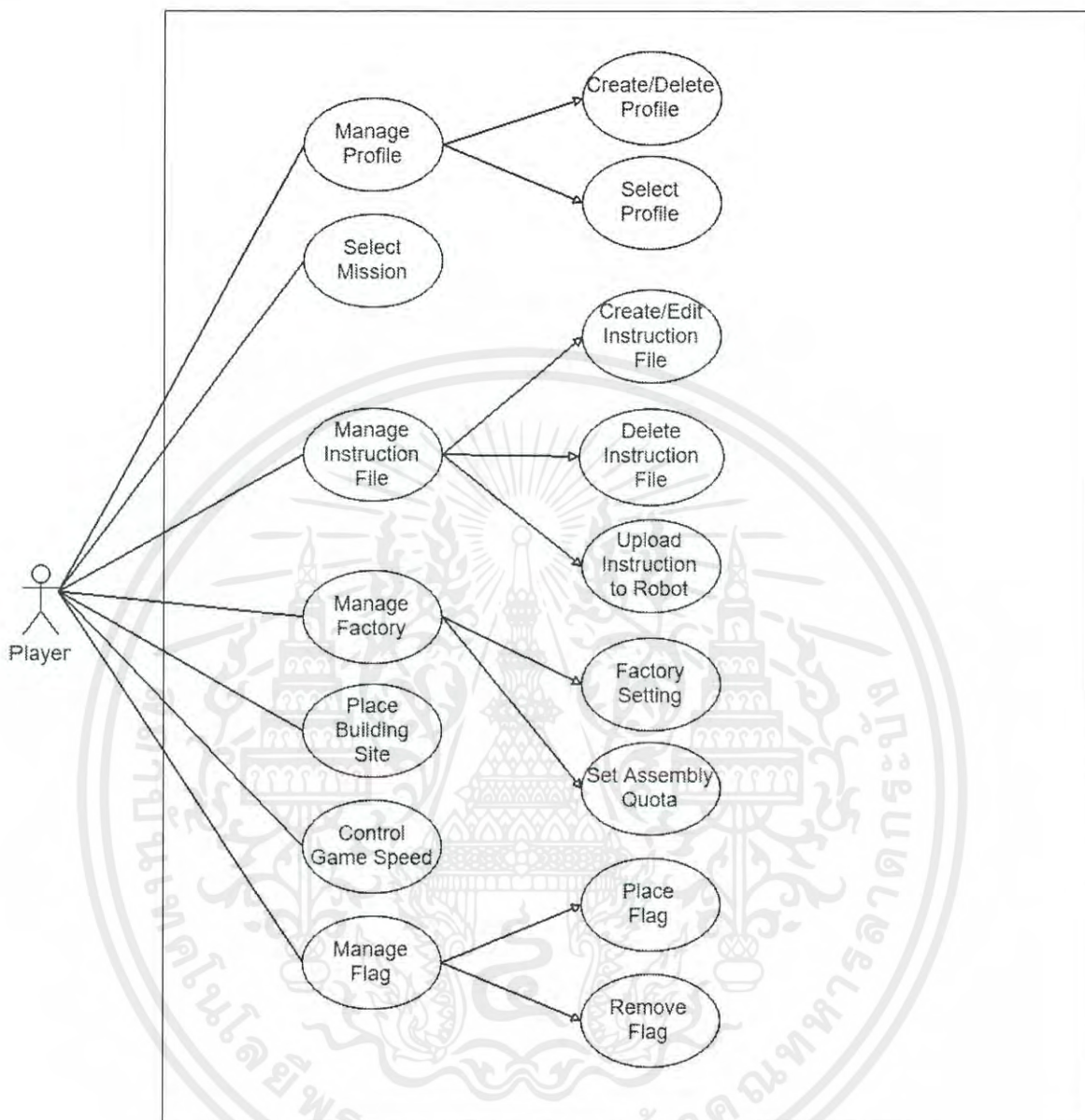
ตาราง 3.7 User requirement specification

Details	Type	Priority
สามารถเขียนคำสั่งลงหุ้่นได้ และหุ้่นยนต์จะทำตามคำสั่งที่เขียนไว้ตามขั้นตอนโดยอัตโนมัติ	-Gameplay -User Interface -Functional	จำเป็นต้องมี
มีระบบอ้างอิงระบุตำแหน่งให้หุ้่นยนต์ทำงานต่างๆตามที่คำสั่งระบุ	-Gameplay -User Interface -Functional	จำเป็นต้องมี
มีแหล่งทรัพยากรให้เก็บเกี่ยวและใช้งาน	-Gameplay -Functional	จำเป็นต้องมี

ตาราง 3.7 User requirement specification (ต่อ)

มีระบบการก่อสร้างตึกประเภทต่างๆ	-Gameplay -Functional	จำเป็นต้องมี
มีระบบการประกอบหุ่นยนต์ประเภทต่างๆ	-Gameplay -Functional	จำเป็นต้องมี
มีการต่อสู้กันระหว่างหุ่นยนต์สองฝ่าย	-Gameplay -Function	จำเป็นต้องมี
มีระบบการทำภารกิจประเภทต่างๆ	-Gameplay -Goal -Function	จำเป็นต้องมี
สามารถคุมเวลาเกมได้	-Gameplay -User Interface -Functional	จำเป็นต้องมี
มีระบบโปรไฟล์ โดยแต่ละโปรไฟล์จะเก็บข้อมูลของแต่ละผู้เล่นแยกออกจากกัน	-System -Game Data -Functional	จำเป็นต้องมี
มุมมองการเล่นเป็น 3 มิติ	-User Interface -Functional	จำเป็นต้องมี
มีเนื้อเรื่องและ Cutscene	-Story -Non-functional	ควรมี
เสียงประกอบเกมและ Animation	-User Interface -Non-functional	ควรมี
ข้อความแสดงการเปลี่ยนแปลงและสถานการณ์	-User Interface -Non-functional	ควรมี

3.6 Use case diagram



รูป 3.1 use case diagram

ตาราง 3.8 รายละเอียดของ Use case

Use case	คำอธิบาย
Manage Profile	จัดการโปรไฟล์ของผู้เล่น โดยแต่ละโปรไฟล์จะแยกความคืบหน้า ไฟล์ คำสั่ง และสถิติเกมของแต่ละผู้เล่นออกจากกัน
Select Mission	เลือกภารกิจที่จะเล่น โดยจะอิงความคืบหน้าจากโปรไฟล์ที่เลือกไว้
Manage Instruction File	จัดการไฟล์คำสั่งภายในเกม สร้าง แก้ไข บันทึก ลบ และอัปโหลดไฟล์ ลงหุ่นยนต์หรือโรงงานประกอบหุ่นได้
Manage Factory	ตั้งค่าการประกอบหุ่นของโรงงาน เลือกประเภทหุ่นที่จะประกอบและ ไฟล์คำสั่งที่จะอัปโหลดลงหุ่น
Place Building Site	วางแปลนสิ่งก่อสร้าง
Manage Flag	จัดการธงอ้างอิง โดยสามารถปักใหม่ เคลื่อนย้าย หรือถอนออกได้
Control Game Speed	ควบคุมความเร็วภายในเกม

ตาราง 3.9 รายละเอียดของ Actor

Actor	คำอธิบาย
Player	ผู้เล่นเกม

3.6.1 Fully Dressed Use case

ตาราง 3.10 รายละเอียดของ Use Case สร้างบัญชีผู้เล่น

Use case: Create Profile
Use case ID:
Actors: Player
Preconditions: ไม่มี
Flow of events: <ol style="list-style-type: none"> 1. ในหน้าเลือกโปรไฟล์ กดปุ่ม New Profile 2. ใส่ชื่อโปรไฟล์ แล้วกด Create
Post conditions: <ol style="list-style-type: none"> 1. ระบบจะสร้างโปรไฟล์ว่างเปล่าที่มีชื่อตามที่ระบุ

ตาราง 3.11 รายละเอียดของ Use Case เลือกบัญชีผู้เล่น

Use case: Select Profile
Use case ID:
Actors: Player
Preconditions: <ol style="list-style-type: none"> 1. ต้องมีโปรไฟล์ในระบบอย่างน้อย 1 โปรไฟล์
Flow of events: <ol style="list-style-type: none"> 1. ในหน้าเลือกโปรไฟล์ กดเลือกโปรไฟล์ที่จะใช้ 2. กดปุ่ม Select
Post conditions: <ol style="list-style-type: none"> 1. ระบบจะอ้างอิงและบันทึกความคืบหน้าและข้อมูลต่างๆลงโปรไฟล์ที่เลือก

ตาราง 3.12 รายละเอียดของ Use Case ลบบัญชีผู้เล่น

Use case: Delete Profile
Use case ID:
Actors: Player
Preconditions: <ol style="list-style-type: none"> 1. ต้องมีโปรไฟล์ในระบบอย่างน้อย 1 โปรไฟล์
Flow of events: <ol style="list-style-type: none"> 1. ในหน้าเลือกโปรไฟล์ กดเลือกโปรไฟล์ที่จะลบ 2. กด Delete จะมีหน้าต่างยืนยัน กด Delete อีกครั้ง
Post conditions: <ol style="list-style-type: none"> 1. ระบบจะลบโปรไฟล์ที่เลือกออกจากระบบ

ตาราง 3.13 รายละเอียดของ Use Case เลือกด่าน

Use case: Select Mission
Use case ID:
Actors: Player
Preconditions: 1. ต้องตั้งค่าโปรไฟล์ที่จะใช้ก่อน (“Select Profile” Use case)
Flow of events: 1. ในหน้าเมนูหลัก กดปุ่ม Mission 2. ในหน้าเลือกภารกิจ กดเลือกภารกิจที่จะเล่น 3. กด Start Mission
Post conditions: 1. ระบบจะโหลดข้อมูลภารกิจที่เลือก

ตาราง 3.14 รายละเอียดของ Use Case สร้างไฟล์คำสั่ง

Use case: Create Instruction File
Use case ID:
Actors: Player
Preconditions: ไม่มี
Flow of events: 1. ภายในเกม กดปุ่ม Instruction ตรงแถบด้านขวาของจอ 2. จะมีหน้าต่างจัดการไฟล์คำสั่งขึ้นมา กดปุ่ม New 3. หน้าต่างเขียนคำสั่งว่างเปล่าปรากฏขึ้น จะมีบล็อกคำสั่งต่างๆ ให้เลือกใช้ 4. เมื่อเขียนคำสั่งเสร็จ ตั้งชื่อแล้ว กดปุ่ม Save 5. ถ้าชื่อที่ตั้งซ้ำกับไฟล์เดิมที่มีอยู่ก่อนหน้า จะมีหน้าต่างให้ยืนยันการเขียนทับ
Post conditions: 1. ระบบจะสร้าง (หรือเขียนทับ) ไฟล์คำสั่งตามชื่อที่ตั้งไว้

ตาราง 3.15 รายละเอียดของ Use Case แก้ไขไฟล์คำสั่ง

Use case: Edit Instruction File
Use case ID:
Actors: Player
Preconditions: <ol style="list-style-type: none"> 1. ต้องมีไฟล์คำสั่งในโปรไฟล์ของผู้เล่นอย่างน้อย 1 ไฟล์
Flow of events: <ol style="list-style-type: none"> 1. ภายในเกม กดปุ่ม Instruction ตรงแถบด้านขวาของจอ 2. จะมีหน้าต่างจัดการไฟล์คำสั่งขึ้นมา เลือกไฟล์คำสั่งแล้วกดปุ่ม Edit 3. หน้าต่างเขียนคำสั่งที่มีข้อมูลปรากฏขึ้น 4. เมื่อแก้ไขไฟล์คำสั่งเสร็จ กดปุ่ม Save 5. จะมีหน้าต่างให้ยืนยันการเขียนทับ (ในกรณีคงชื่อเดิมไว้)
Post conditions: <ol style="list-style-type: none"> 1. ระบบจะสร้าง (หรือเขียนทับ) ไฟล์คำสั่งตามชื่อที่ตั้งไว้

ตาราง 3.16 รายละเอียดของ Use Case ลบไฟล์คำสั่ง

Use case: Delete Instruction File
Use case ID:
Actors: Player
Preconditions: <ol style="list-style-type: none"> 1. ต้องมีไฟล์คำสั่งในโปรไฟล์ของผู้เล่นอย่างน้อย 1 ไฟล์
Flow of events: <ol style="list-style-type: none"> 1. ภายในเกม กดปุ่ม Instruction ตรงแถบด้านขวาของจอ 2. จะมีหน้าต่างจัดการไฟล์คำสั่งขึ้นมา เลือกไฟล์คำสั่งแล้วกด Delete 3. จะมีหน้าต่างยืนยัน กด Delete อีกครั้ง
Post conditions: <ol style="list-style-type: none"> 1. ระบบจะลบไฟล์คำสั่งนั้นออกจากระบบ

ตาราง 3.17 รายละเอียดของ Use Case นำไฟล์คำสั่งเข้าหุ่นยนต์

Use case: Upload Instruction File to Robot
Use case ID:
Actors: Player
Preconditions: <ol style="list-style-type: none"> 1. ต้องมีไฟล์คำสั่งในโปรไฟล์ของผู้เล่นอย่างน้อย 1 ไฟล์ 2. ต้องมีหุ่นยนต์อย่างน้อย 1 ตัว
Flow of events: <ol style="list-style-type: none"> 1. ภายในเกม เลือกหุ่นยนต์ที่จะอัปโหลดไฟล์คำสั่ง 2. กดปุ่ม Instruction ตรงแถบด้านขวาของจอ 2. จะมีหน้าต่างจัดการไฟล์คำสั่งขึ้นมา เลือกไฟล์คำสั่งแล้วกดปุ่ม Upload
Post conditions: <ol style="list-style-type: none"> 1. หุ่นที่เลือกจะทำการรีบูตแล้วเริ่มทำตามคำสั่งที่อัปโหลดใหม่

ตาราง 3.18 รายละเอียดของ Use Case การตั้งค่าโรงงาน

Use case: Factory Setting
Use case ID:
Actors: Player
Preconditions: <ol style="list-style-type: none"> 1. ต้องมีไฟล์คำสั่งในโปรไฟล์ของผู้เล่นอย่างน้อย 1 ไฟล์ 2. ต้องมีโรงงานอย่างน้อย 1 ศึก
Flow of events: <ol style="list-style-type: none"> 1. ภายในเกม เลือกโรงงานที่จะทำการตั้งค่า 2. ตรงกรอบข้อมูลมุมขวาล่างจอ จะปรากฏรายละเอียดของโรงงานที่เลือก 3. กดปุ่ม Setting แล้วจะมีหน้าต่างตั้งค่าโรงงานขึ้นมา 4. เลือกแบบหุ่นที่จะประกอบ พร้อมทั้งไฟล์คำสั่งที่ต้องการใช้ แล้วกดปุ่ม Set
Post conditions: <ol style="list-style-type: none"> 1. โรงงานที่เลือกจะประกอบหุ่นที่ระบุ และหุ่นที่ออกมาจะทำตามคำสั่งที่ใช้

ตาราง 3.19 รายละเอียดของ Use Case เลือกจำนวนหุ่นยนต์ที่จะสร้าง

Use case: Set Assembly Quota
Use case ID:
Actors: Player
Preconditions: 1. ต้องมีโรงงานที่ตั้งค่าไว้แล้วอย่างน้อย 1 คีค (“Factory Setting” Use case)
Flow of events: 1. ภายในเกม เลือก โรงงานที่จะสั่งประกอบ 2. ตรวจสอบข้อมูลมุมขวาล่างจอ จะปรากฏรายละเอียดของโรงงานที่เลือก 3. ระบุจำนวนหุ่นที่จะประกอบ
Post conditions: 1. โรงงานที่เลือกจะประกอบหุ่นตามที่ตั้งไว้ ตามจำนวนที่ระบุ เมื่อมีทรัพยากรพอ

ตาราง 3.20 รายละเอียดของ Use Case วางสิ่งก่อสร้าง

Use case: Place Building Site
Use case ID:
Actors: Player
Preconditions: ไม่มี
Flow of events: 1. ภายในเกม กดปุ่ม Build ตรงแถบด้านขวาของจอ 2. จะปรากฏสิ่งก่อสร้างที่สามารถสร้างได้ 3. กดเลือกสิ่งก่อสร้างแล้วระบุตำแหน่งภายในเกม
Post conditions: 1. จะมีเขตก่อสร้างปรากฏขึ้นมา ให้หุ่นมาทำการก่อสร้าง

ตาราง 3.21 รายละเอียดของ Use Case ควบคุมความเร็วเกม

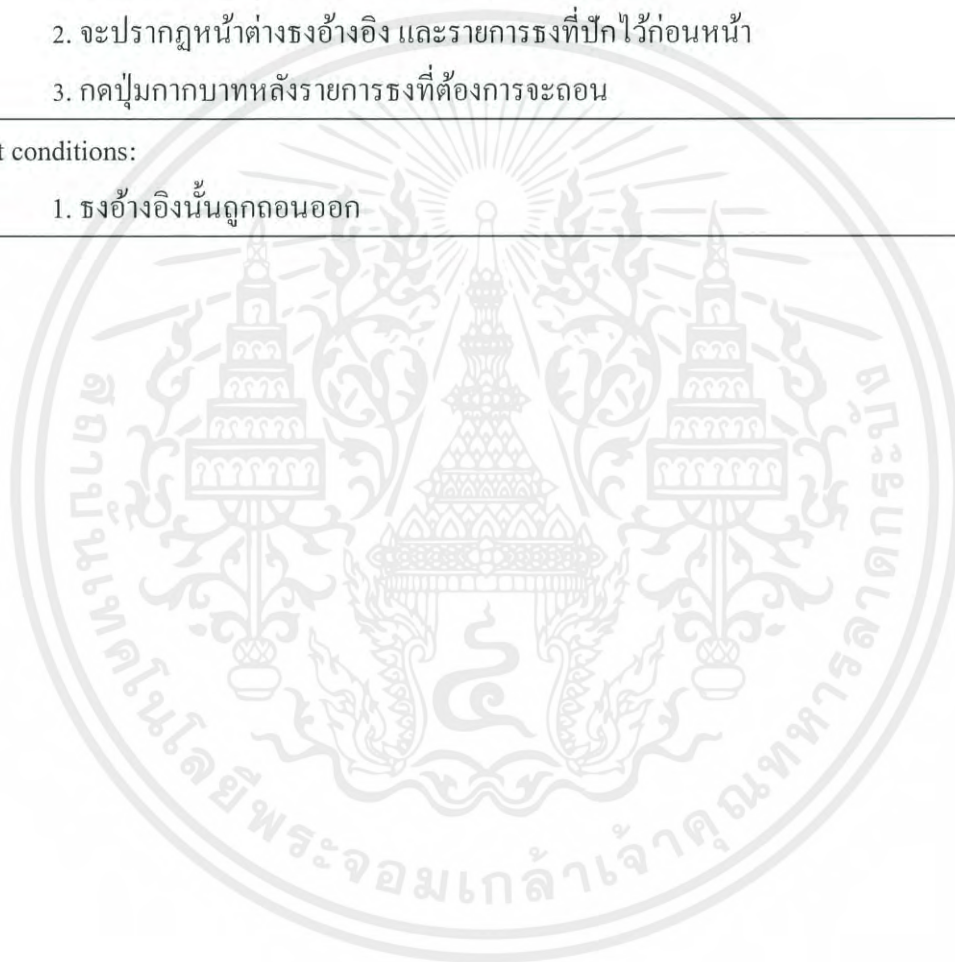
Use case: Control Game Speed
Use case ID:
Actors: Player
Preconditions: ไม่มี
Flow of events: <ol style="list-style-type: none"> 1. ภายในเกม จะมีปุ่มควบคุมความเร็วเกมมุมซ้ายล่าง และข้อมูลความเร็วเกม - กดปุ่ม Faster หรือ Slower เพื่อเปลี่ยนความเร็วเกม - กดปุ่ม Play/Pause เพื่อหยุดเกมชั่วคราว หรือให้เกมดำเนินต่อ - กดปุ่ม Step เมื่อเกมหยุดเดิน เพื่อให้เกมดำเนินไป 1 ขั้นตอน
Post conditions: <ol style="list-style-type: none"> 1. ความเร็วการดำเนินเกมเปลี่ยนแปลง

ตาราง 3.22 รายละเอียดของ Use Case วางและย้ายธง

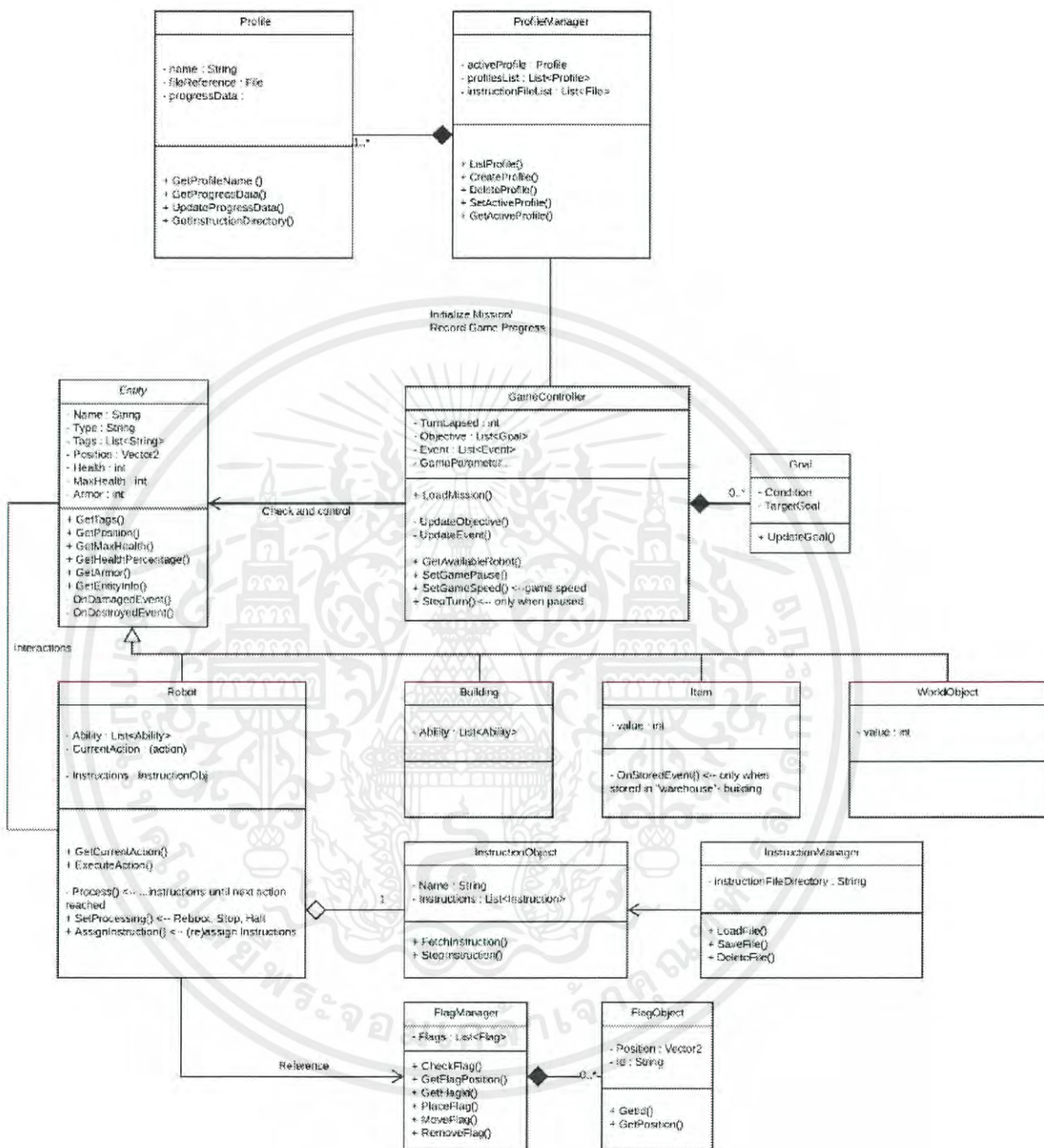
Use case: Place or Move Flag
Use case ID:
Actors: Player
Preconditions: ไม่มี
Flow of events: <ol style="list-style-type: none"> 1. ภายในเกม กดปุ่ม Flag ตรงแถบด้านขวาของจอ 2. จะปรากฏหน้าต่างธงอ้างอิง และรายการธงที่ปักไว้ก่อนหน้านี้ (ถ้ามี) 3. กดเลือกสัญลักษณ์ของธงแล้วระบุตำแหน่งภายในเกม
Post conditions: <ol style="list-style-type: none"> 1. จะมีธงอ้างอิงที่มีสัญลักษณ์ที่ระบุปักไว้ ณ ตำแหน่งนั้น 2. ถ้าเป็นธงสัญลักษณ์ที่ซ้ำกับของเดิมที่มีก่อนหน้านี้จะเป็นการย้ายธงแทน

ตาราง 3.23 รายละเอียดของ Use Case ลบธง

Use case: Remove Flag
Use case ID:
Actors: Player
Preconditions: <ol style="list-style-type: none"> 1. ต้องมีธงอ้างอิงปักไว้แล้ว
Flow of events: <ol style="list-style-type: none"> 1. ภายในเกม กดปุ่ม Flag ตรงแถบด้านขวาของจอ 2. จะปรากฏหน้าต่างธงอ้างอิง และรายการธงที่ปักไว้ก่อนหน้า 3. กดปุ่มกากบาทหลังรายการธงที่ต้องการจะถอน
Post conditions: <ol style="list-style-type: none"> 1. ธงอ้างอิงนั้นถูกถอนออก

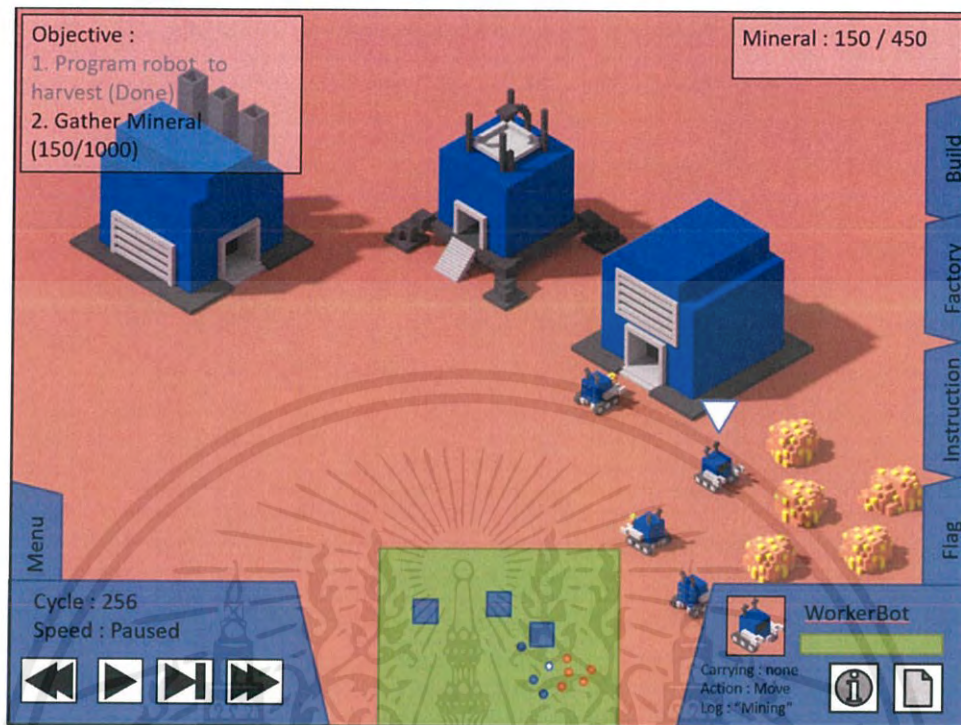


3.7 Class Diagram

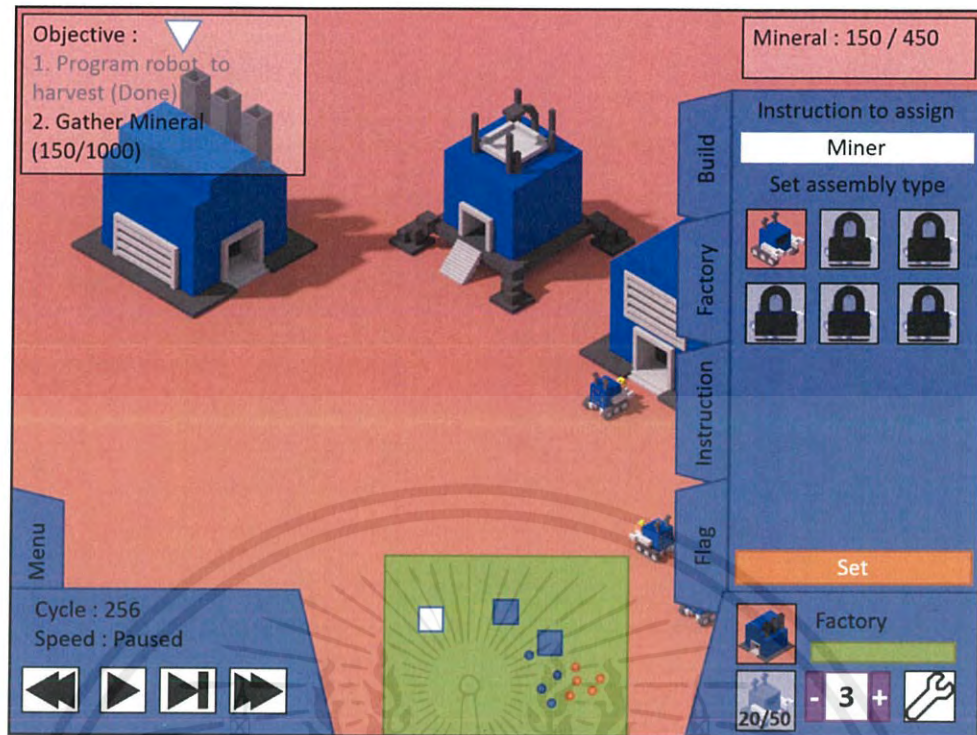


รูป 3.2 class diagram

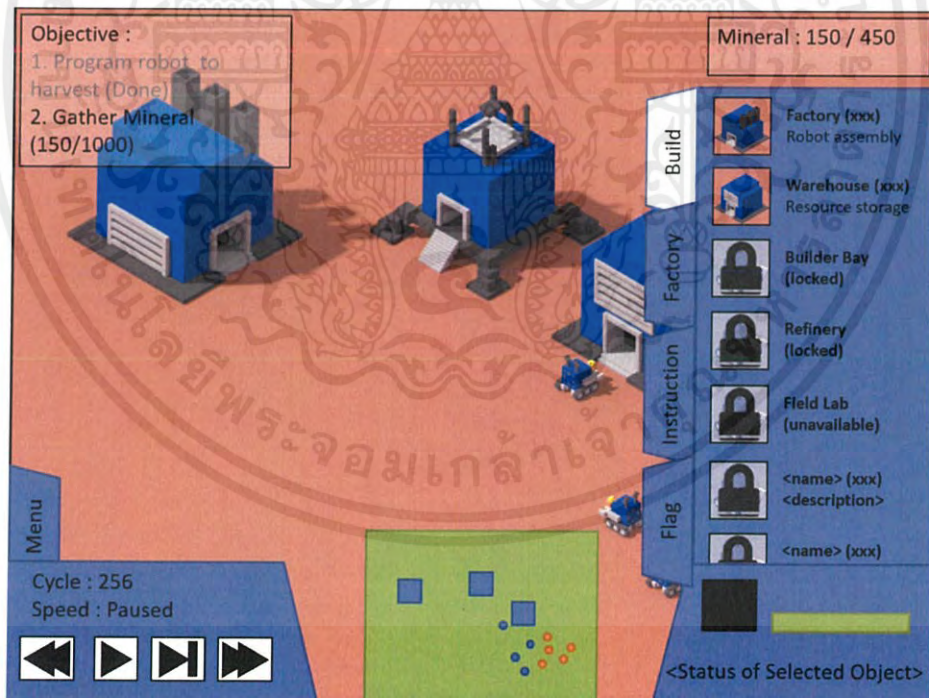
3.8 การออกแบบหน้าต่างโปรแกรม



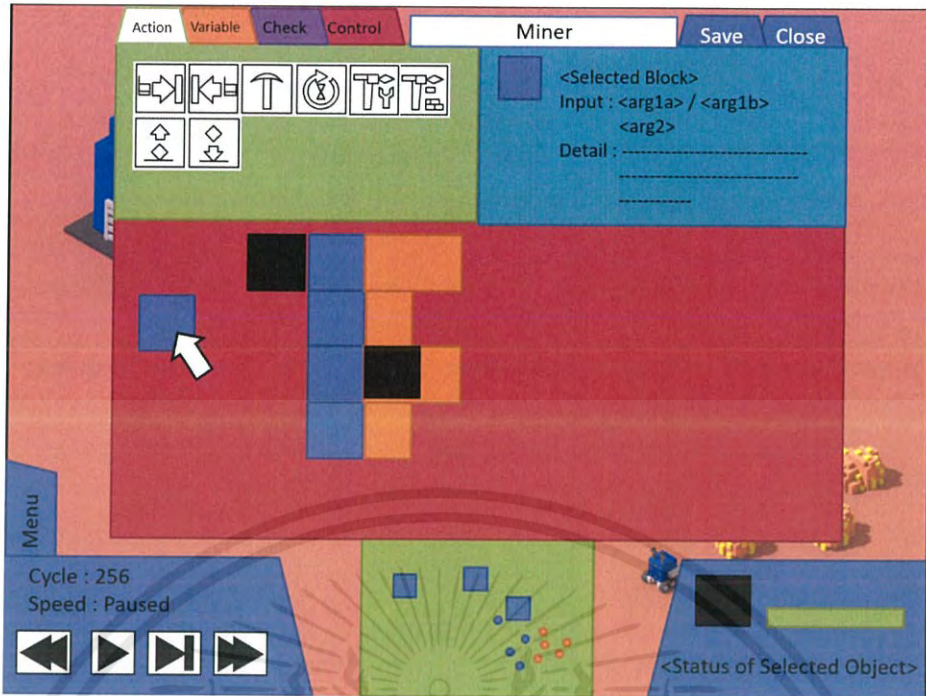
รูป 3.3 หน้าตาโปรแกรมเมื่อคลิกเลือกยูนิท



รูป 3.4 หน้าตาโปรแกรมเมื่อกดเลือกโรงประกอบหุ่นยนต์



รูป 3.5 หน้าตาโปรแกรมเมื่อสร้างสิ่งก่อสร้าง



รูป 3.6 หน้าตาโปรแกรมเมื่อเขียนโปรแกรมคำสั่ง

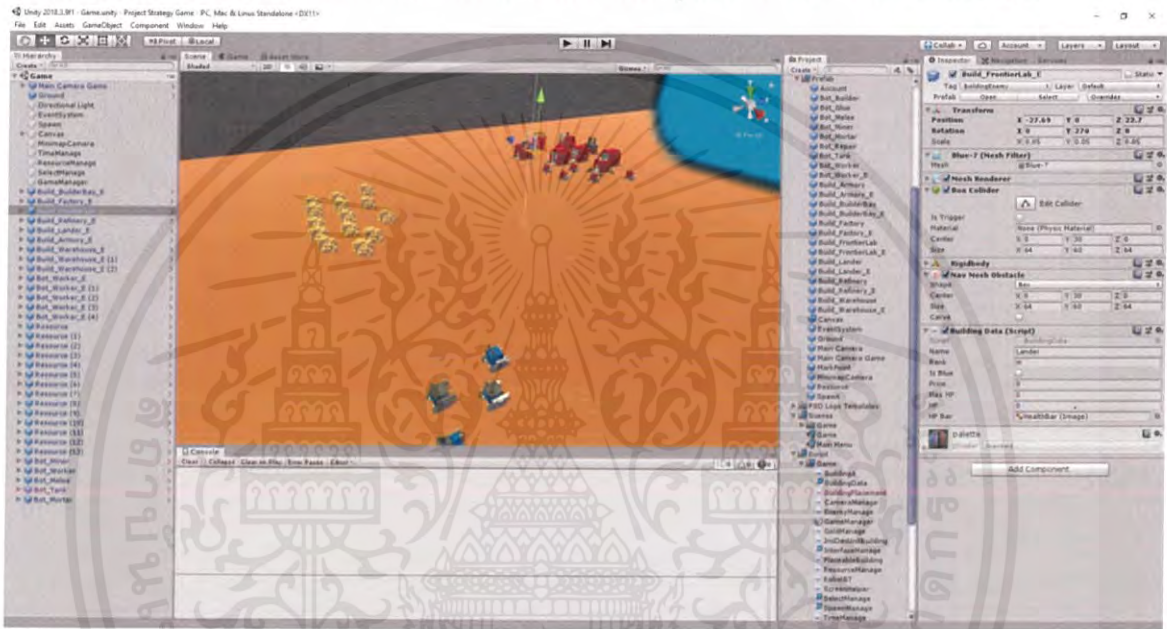
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การพัฒนาเกม

ในการพัฒนาโปรแกรมเกมนี้ ได้ใช้โปรแกรมเกมเอนจินของ Unity3D มาช่วยในการจัดการองค์ประกอบส่วนต่างๆ ภายในของเกมนั้นและทำการเขียนโปรแกรมให้กับวัตถุต่างๆ ภายในเกมให้ทำงานออกมาได้ตามที่ต้องการ โดยตัวโปรแกรมเกมในปัจจุบันสามารถทำงานได้ดังต่อไปนี้



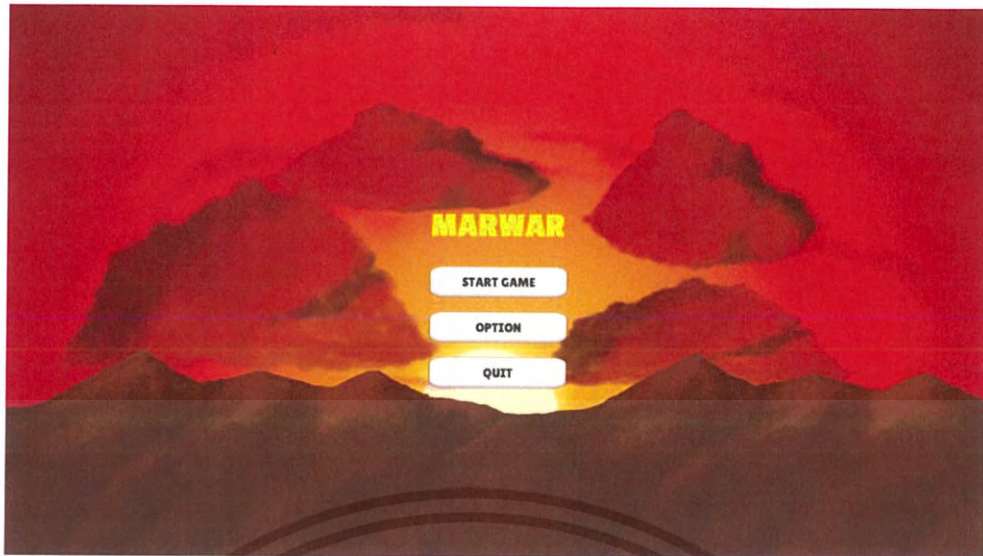
รูป 4.1 โปรแกรม Unity3D

4.2 หน้าเมนูหลัก

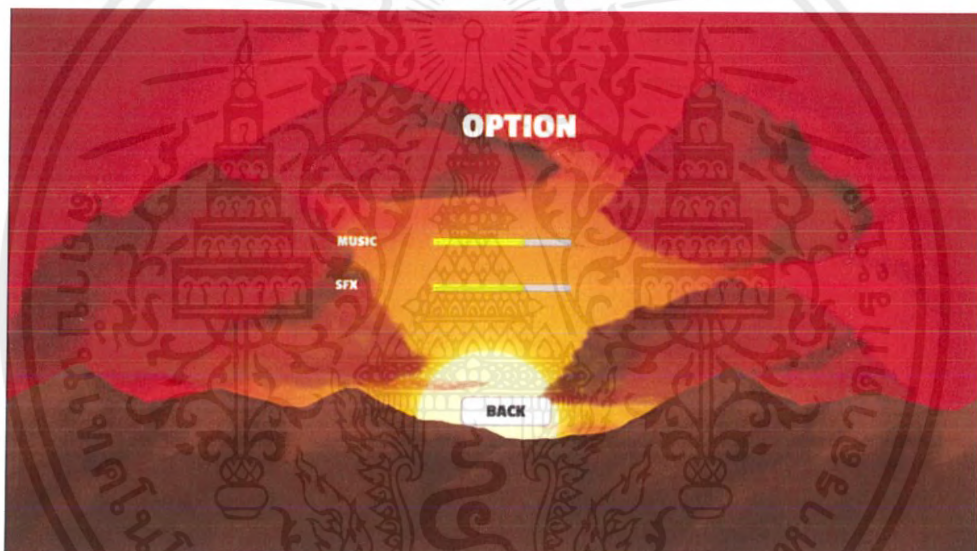
ในส่วนนี้เป็นส่วนแรกๆที่ผู้เล่นจะได้พบตั้งแต่เปิดโปรแกรมเกม เป็นส่วนที่จะเชื่อมโยงไปหาส่วนต่างๆของโปรแกรม โดยจะมีปุ่มกด 3 ปุ่มได้แก่

- 1) Start Game ปุ่มสำหรับเข้าไปหน้าต่างเลือกด่านภายในเกม
- 2) Option ปุ่มสำหรับตั้งค่าต่างภายในเกม ได้แก่ ความดังของเสียงเพลงและเสียงองค์ประกอบต่างๆภายในเกม
- 3) Quit ปุ่มเพื่อใช้ในการออกจากโปรแกรม

เมื่อกดปุ่ม Start Game จะเข้าสู่หน้าต่างเลือกด่านต่างๆของเกม ซึ่งมีทั้งหมด 5 ด่าน โดยปัจจุบันสามารถกดด่านที่ 1 ได้ เมื่อกดปุ่มด่าน 1 จะเข้าสู่เกมทันที



รูป 4.2 หน้าต่างเมนูหลัก



รูป 4.3 หน้าต่าง Option

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.4 หน้าต่างเลือกด่านภายในเกม

4.3 การเคลื่อนที่ของกล้อง

ภายในเกม เราจะเล่นเกมในมุมมองมองจากด้านบนผ่านกล้องที่มองลงมา เราสามารถใช้เมาส์เคลื่อนไปอยู่ตามขอบมุมของจอภาพหรือคีย์บอร์ด W A S D ของคีย์บอร์ดเพื่อใช้ในการบังคับกล้องให้เคลื่อนที่ไปอยู่ในจุดที่เราต้องการได้ นอกจากนี้สามารถบังคับให้กล้องซูมเข้าหรือออกโดยใช้ Scroll bar ของเมาส์ได้

4.4 ระบบเงิน

ภายในเกมผู้เล่นสามารถสร้างสิ่งก่อสร้างและหุ่นยนต์ประเภทต่างๆ ได้โดยใช้เงิน (Gold) ตามจำนวนที่กำหนดไว้ รายละเอียดของราคาส่งก่อสร้างและหุ่นยนต์ดูได้ในบทที่ 3 จำนวนเงินสามารถดูได้จากมุมซ้ายบนของเกม การได้ซึ่งเงินภายในเกมสามารถได้มาด้วย 2 วิธีคือจากการเพิ่มขึ้นเองอัตโนมัติ แต่จะได้มาจำนวนน้อย และจากการขุดแร่โดยใช้หุ่นยนต์ที่มีความสามารถในการขุดแร่ เช่น Miner Bot หรือ Worker Bot ซึ่งวิธีนี้เป็นวิธีหาเงินมาได้จำนวนมากที่สุด



รูป 4.5 หน้าต่างแสดงจำนวนเงินที่มีอยู่

ตัวเลขจำนวนเงินจะมี 2 ส่วนคือจำนวนเงินที่มีอยู่ในปัจจุบันและจำนวนเงินที่สามารถเก็บได้ทั้งหมด ซึ่งมี / เป็นตัวคั่นทั้ง 2 ส่วน โดยจำนวนเงินที่มีอยู่ปัจจุบันจะไม่สามารถมากกว่าจำนวนเงินที่เก็บได้ แต่จำนวนเงินที่สามารถเก็บได้สามารถเพิ่มขีดจำกัดในการเก็บเงินได้โดยการสร้างสิ่งก่อสร้างประเภท Warehouse โดย 1 Warehouse จะเพิ่มขีดจำกัดในการเก็บเงินได้ 100 หน่วย



รูป 4.6 จำนวนขีดจำกัดในการเก็บเงินเมื่อยังไม่มี Warehouse



รูป 4.7 จำนวนขีดจำกัดในการเก็บเงินเมื่อมี Warehouse

4.5 Minimap

Minimap เป็นแผนที่ที่ใช้ในการมองภาพรวมของเกมว่ามีสิ่งใดอยู่ภายในเกมด้านๆนั้นบ้าง จะอยู่ตรงกลางด้านล่างของหน้าจอเกม และมีกรอบสีเขียวที่ใช้ในการระบุว่าขณะนี้เรากำลังมองอยู่ส่วนไหนของเกมอยู่ ซึ่งกรอบสีเขียวจะเคลื่อนไปตามตำแหน่งของกล้องที่เราเคลื่อนที่ไป



รูป 4.8 หน้าต่าง Minimap

4.6 แหล่งทรัพยากร

จะเป็นก๊องแร่อยู่ตามด่าน เมื่อนำหุ่นยนต์ที่มีความสามารถในการขุดแร่เช่น Miner Bot หรือ Worker Bot ไปสั่งให้ขุดแร่ โดยเมื่อเริ่มขุดแร่ ที่ตัวแร่จะมีหลอดเลือดของแร่ปรากฏขึ้นมาบอกสถานะของแร่ก้อนนั้น เมื่อหลอดเลือดของแร่ก้อนนั้นหมดลงไป ก๊องแร่นั้นจะหายไปและหุ่นยนต์จะขุดแร่เสร็จสิ้น จะเป็นการเพิ่มจำนวนเงินของเราที่มี โดยแร่ 1 ก้อนจะให้เงินได้ 20 หน่วย



รูป 4.9 หุ่นยนต์ Miner กำลังขุดแร่

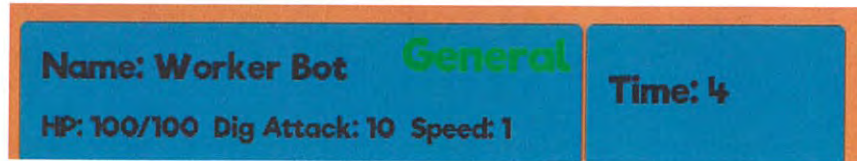
4.7 แถบแสดงสถานะ

ด้านล่างมุมซ้ายของหน้าจอเกมจะมีแถบแสดงสถานะของสิ่งก่อสร้างหรือหุ่นยนต์ของเรา เมื่อเราคลิกซ้ายไปที่สิ่งก่อสร้างหรือหุ่นยนต์ของเราจะมีข้อมูลต่างปรากฏขึ้นมา

เมื่อเราคลิกเลือกหุ่นยนต์ แถบแสดงสถานะจะแสดงชื่อ พลังชีวิต และประเภทของหุ่นยนต์ขึ้นมา ซึ่งหุ่นยนต์แต่ละประเภทจะแสดงสถานะแตกต่างกันไป โดยประเภท Attack จะแสดงพลังโจมตี และความเร็วในการโจมตี และประเภท General จะแสดงพลังในการขุดแร่และความเร็วในการขุดแร่



รูป 4.10 แถบสถานะเมื่อเลือกหุ่นยนต์ประเภท Attack



รูป 4.11 แถบสถานะเมื่อเลือกหุ่นยนต์ประเภท General

เมื่อคลิกเลือกสิ่งก่อสร้าง แถบแสดงสถานะจะแสดงชื่อ พลังชีวิตของสิ่งก่อสร้าง และจะมีปุ่มกากบาทสีแดง เมื่อกดปุ่มแล้วจะทำการทำลายสิ่งก่อสร้างของเราที่ได้เลือกไว้



รูป 4.12 แถบสถานะเมื่อเลือกที่สิ่งก่อสร้าง

4.8 การหยุดพักเกม

ผู้เล่นสามารถหยุดเกมได้ชั่วคราว โดยกดปุ่มด้านมุมบนซ้ายของเกม เมื่อกดปุ่มจะทำให้โปรแกรมเกมหยุดนิ่ง พร้อมขึ้นหน้าจอให้เลือกว่าจะเริ่มเล่นต่อหรือกลับไปหน้าจอหลักของเกม



รูป 4.13 การหยุดพักเกม

4.9 การสร้างสิ่งก่อสร้าง

สิ่งก่อสร้างภายในเกมมีหลายประเภท และแต่ละประเภทมีความสามารถและจำนวนเงินที่ใช้ในการสร้างแตกต่างกัน โดยสามารถสร้างได้จากหน้าต่างด้านขวาของเกมในหมวด Building และจำนวนปั้มของสิ่งก่อสร้างประเภทต่างๆ จะเพิ่มขึ้นเมื่อทำการสร้างสิ่งก่อสร้าง Builder Bay ขึ้นมา

การสร้างสิ่งก่อสร้างทำโดยการคลิกซ้ายปั้มของสิ่งก่อสร้างนั้น จากนั้นจะแสดงตำแหน่งของสิ่งก่อสร้างขึ้นมาเพื่อใช้ในการวางตำแหน่งของสิ่งก่อสร้างตามตำแหน่งของเม้าส์ เมื่อคลิกซ้ายอีกทีจะเป็นการเลือกตกลงที่จะสร้างสิ่งก่อสร้างตามตำแหน่งที่ได้เลือกไว้ แต่หากคลิกขวาจะเป็นการยกเลิกการสร้างสิ่งก่อสร้างนั้น

เงื่อนไขในการสร้างสิ่งก่อสร้างแต่ละประเภทจะแตกต่างกันไป บางประเภทจะสามารถสร้างได้เพียง 1 หลังเท่านั้น เช่น Builder Bay, Armory เป็นต้น และเมื่อสร้างสิ่งก่อสร้างแล้วจะเป็นการปลดล็อคหุ่นยนต์ประเภทต่างๆ ให้สามารถสร้างได้ อาทิเช่น เมื่อสร้าง Armory แล้วจะสามารถสร้างหุ่นยนต์โจมตีประเภทต่างๆ เพิ่มได้ เช่น Mortar Tank เป็นต้น



รูป 4.14 หน้าต่างการสร้างสิ่งก่อสร้าง



รูป 4.15 การสร้างสิ่งก่อสร้าง

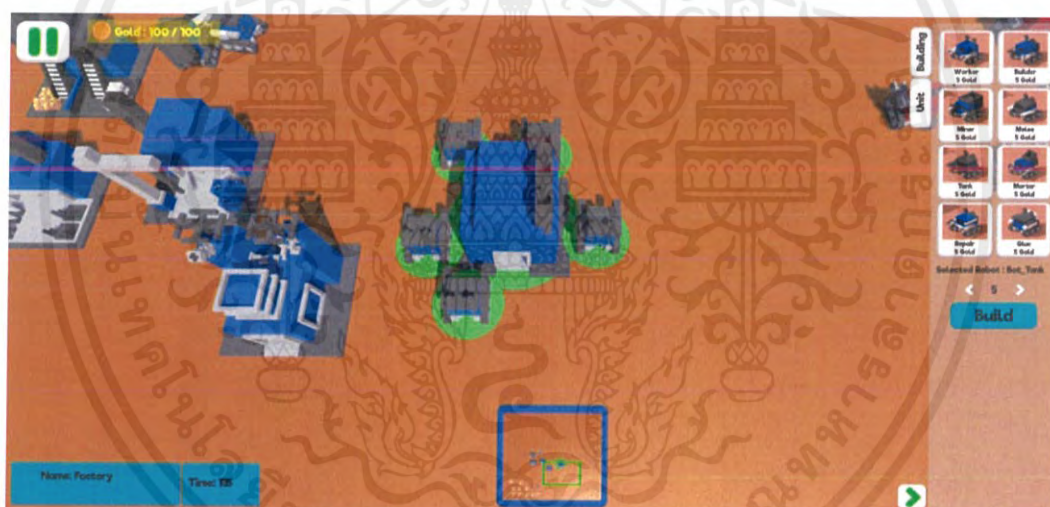
4.10 การสร้างหุ่นยนต์

หุ่นยนต์ภายในเกมจะเป็นตัวละครที่ผู้เล่นสามารถควบคุมได้และผู้เล่นต้องทำการสร้างขึ้นมาเพื่อใช้งานในด้านการขุดแร่เพื่อหาเงินมาใช้และด้านการต่อสู้กับศัตรู โดยหุ่นยนต์แต่ละประเภทจะมีเงื่อนไขในการสร้างและจำนวนเงินที่ต้องใช้ในการสร้างที่แตกต่างกัน

ในส่วนการสร้างสามารถสร้างได้จากสิ่งก่อสร้างประเภท Factory โดยทำการคลิกซ้ายที่สิ่งก่อสร้าง จะมีหน้าต่างด้านขวาขึ้นมาในหมวดของ Unit ทำการคลิกเลือกประเภทของหุ่นยนต์ที่จะสร้างและจำนวนที่ต้องการ จากนั้นกดปุ่ม Build จะทำการสร้างหุ่นยนต์ออกมาตามประเภทและจำนวนที่เลือกไว้ ออกมารอบๆ Factory ที่ได้เลือกไว้ โดยต้องมีจำนวนเงินที่ให้เพียงพอในการสร้าง ถ้าจำนวนเงินไม่เพียงพอจะไม่สามารถกดปุ่ม Build ได้



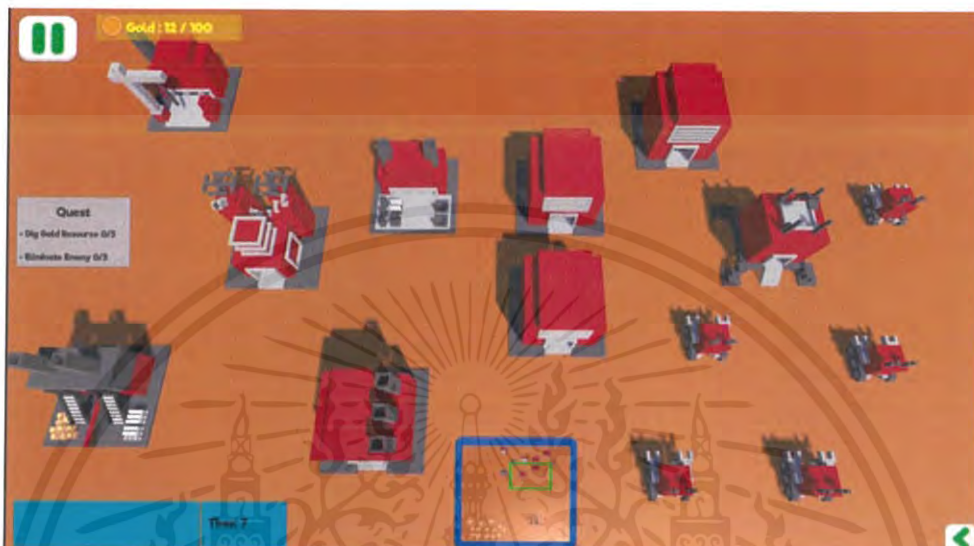
รูป 4.16 หน้าต่างการสร้างหุ่นยนต์



รูป 4.17 หุ่นยนต์ที่สร้างออกมาอยู่ล้อมรอบ Factory

4.11 ศัตรู

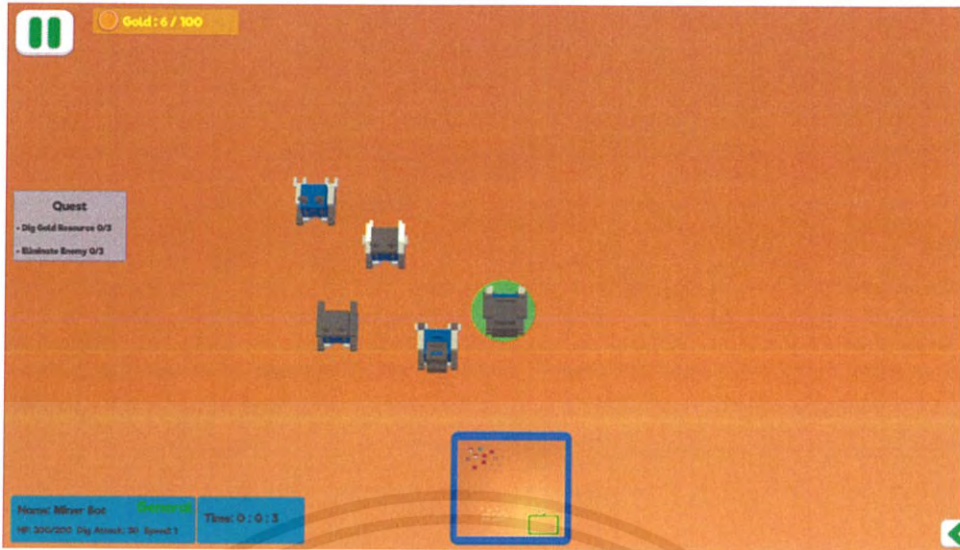
ศัตรูภายในเกมจะประกอบไปด้วยหุ่นยนต์และสิ่งก่อสร้างต่างๆของฝั่งศัตรู โดยจะมีสีแดง เป็นสีธีมหลักของฝั่งศัตรู ทั้งหุ่นยนต์และสิ่งก่อสร้างของศัตรูจะมีหลอดพลังชีวิตอยู่ ถ้าพลังชีวิตหมด วัตถุนั้นจะหายไปจากเกม



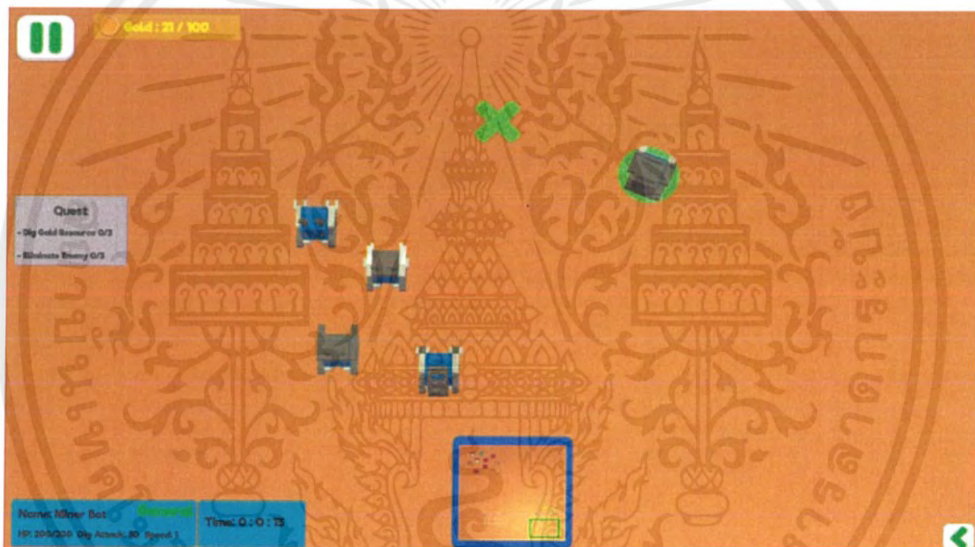
รูป 4.18 หุ่นยนต์และสิ่งก่อสร้างฝั่งศัตรู

4.12 การบังคับตัวละคร

ผู้เล่นสามารถควบคุมตัวละครได้โดยการใช้เมาส์คลิกซ้ายเลือกหุ่นยนต์ จะมีวงกลมสีเขียวปรากฏขึ้นมาใต้หุ่นยนต์ที่เลือก เมื่อทำการคลิกขวาที่พื้นดิน จะเป็นการสั่งให้หุ่นยนต์เคลื่อนที่ไปยังจุดที่ได้คลิกขวาไว้ หากเป็นหุ่นยนต์ที่มีความสามารถในการต่อสู้ เมื่อเลือกหุ่นยนต์ตัวนั้นแล้วคลิกขวาไปที่ศัตรูจะเป็นการสั่งให้หุ่นยนต์ไปโจมตีศัตรู และหากเป็นหุ่นยนต์ที่มีความสามารถในการขุดแร่ เมื่อเลือกหุ่นยนต์แล้วคลิกขวาไปที่ก้อนแร่ จะเป็นการสั่งให้หุ่นยนต์ตัวนั้นไปทำการขุดแร่ และเราสามารถเลือกหุ่นยนต์ได้หลายตัวพร้อมกันได้โดยการคลิกซ้ายค้างแล้วลากเมาส์มาห้กรอบสีเขียวคลุมหุ่นยนต์ที่เราต้องการเลือกได้เช่นกัน

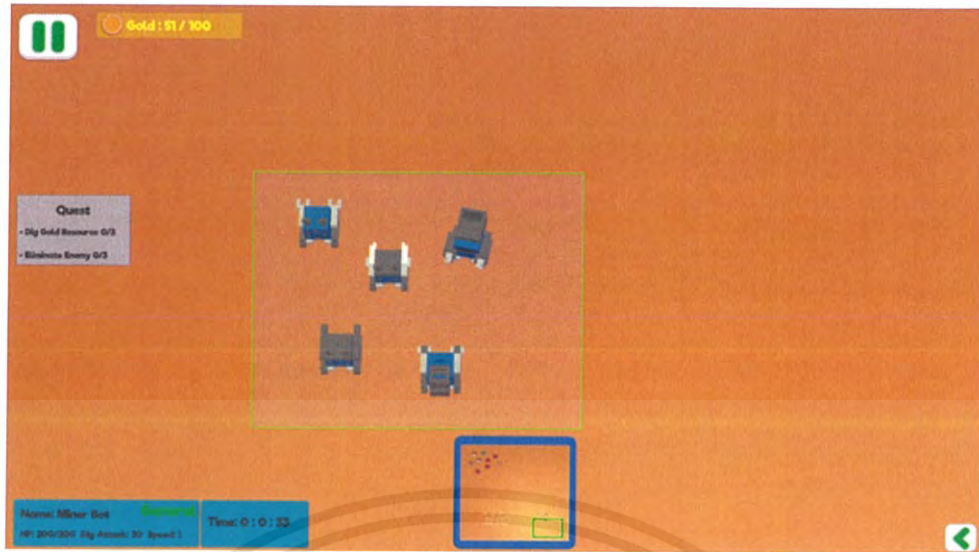


รูป 4.19 การเลือกตัวละคร 1 ตัว



รูป 4.20 การสั่งให้ตัวละครเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



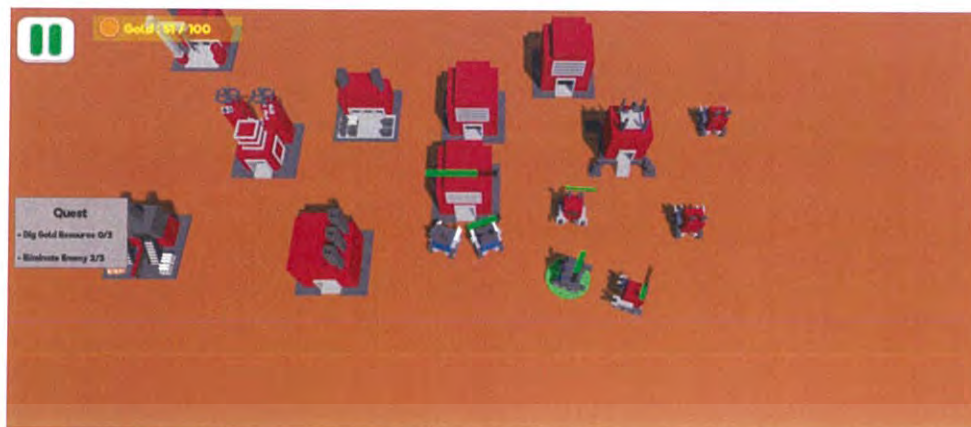
รูป 4.21 การเลือกตัวละครหลายตัว

4.13 การต่อสู้

ผู้เล่นสามารถสั่งให้หุ่นยนต์ของเราทำการโจมตีหุ่นยนต์หรือสิ่งก่อสร้างของฝั่งตรงข้ามได้ เมื่อเกิดการโจมตีกัน วัตถุที่ถูกโจมตีจะแสดงหลอดพลังชีวิตของวัตถุขึ้นมาเหนือวัตถุนั้น และหากหลอดพลังชีวิตของวัตถุนั้นหมด วัตถุนั้นจะถูกทำลายออกจากเกมไปซึ่งจะมีผลทั้งฝั่งผู้เล่นและฝั่งศัตรู



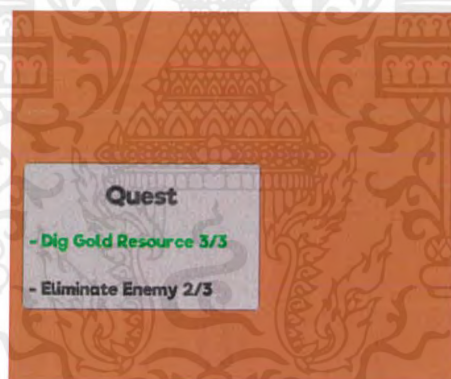
รูป 4.22 การต่อสู้กันระหว่างหุ่นยนต์ทั้ง 2 ฝ่าย



รูป 4.23 หุ่นยนต์ฝึผู้เล่นโจมตีสิ่งก่อสร้างศัตรู

4.14 ภารกิจ

ภายในแต่ละด่านจะมีภารกิจประจำด่าน เป็นเงื่อนไขในการจบด่านๆนั้น โดยภารกิจจะมีแตกต่างกันตามด่าน อาทิเช่น กำจัดศัตรูตามจำนวนที่กำหนด เก็บแร่ให้ได้ตามที่กำหนด หรือทำลายสิ่งก่อสร้างที่กำหนดไว้ เมื่อภารกิจใดสำเร็จอักษรที่บอกภารกิจนั้นจะเปลี่ยนเป็นสีเขียว หากทำทุกภารกิจสำเร็จจะมีหน้าต่างขึ้นมาบอกว่า Mission Complete พร้อมทั้งปุ่มเพื่อกลับไปหน้าเมนูหลัก



รูป 4.24 หน้าต่างภารกิจ



รูป 4.25 หน้าต่างบรรลุมารกิก เมื่อทำภารกิจสำเร็จหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลที่ได้จากการทำโครงการ

- 1) เกมวางแผนกลยุทธ์ที่ให้ความเพลิดเพลินและเสริมสร้างกระบวนการคิดให้แก่ผู้เล่น
- 2) ได้ความรู้และทักษะในการพัฒนาเกมโดยใช้โปรแกรม Unity ช่วยในการออกแบบและพัฒนาเกม อีกทั้งภาษา C# ที่ใช้ในการเขียน Script ให้โปรแกรมเกม
- 3) ได้ฝึกการทำงานเป็นกลุ่มร่วมกันผู้อื่น ฝึกการจัดการเมื่อทำงานเป็นกลุ่ม ฝึกการรับมือและแก้ปัญหาที่เกิดขึ้นระหว่างการทำงาน
- 4) ได้ฝึกทักษะการเรียนรู้ด้วยตนเอง ในการค้นคว้าหาความรู้และนำมาประยุกต์ใช้ในการทำงาน

5.2 ปัญหาและอุปสรรค

เนื่องจากการพัฒนาเกมบนคอมพิวเตอร์โดยใช้โปรแกรม Unity เป็นเรื่องที่คณะผู้จัดทำไม่เคยได้เรียนรู้มาก่อน ทำให้ต้องมีการเรียนรู้ศึกษาการใช้งาน วิธีการใช้งาน เครื่องมือ และความสามารถของโปรแกรมอยู่ตลอดการพัฒนาเกม และในช่วงแรกของการทำงานได้ใช้เวลาในการออกแบบโครงสร้างและองค์ประกอบต่างๆภายในเกมเป็นจำนวนมาก ส่งผลให้การทำงานล่าช้าไปจากแผนงานที่ได้วางไว้

ในช่วงของการพัฒนาโปรแกรมเกมได้เกิดปัญหาขึ้น ตั้งแต่ความยากของการสร้างระบบการเขียนโปรแกรมคำสั่งให้หุ่นยนต์ทำงานได้อัตโนมัติ ทำให้ระบบนี้ยังไม่เสร็จสมบูรณ์ การกำหนดขนาดของงานที่มีขนาดใหญ่เกินไปสำหรับกรอบเวลาที่มีอยู่ ทำให้มีส่วนของโปรแกรมที่ถูกตัดทอนออกไป รวมไปถึงการทำงานร่วมกันของทางผู้จัดทำทั้งด้านการสื่อสารที่ไม่ชัดเจนทำให้เกิดการเข้าใจผิดและด้านการจัดการงานต่างๆ ส่งผลให้ความคืบหน้าของงานช้ากว่าแผนงานที่ได้กำหนดไว้และทำให้ตัวโปรแกรมเกมไม่สามารถทำงานได้สมบูรณ์ตามที่ได้วางแผนไว้

5.3 แนวทางในการพัฒนาต่อ

จากการพิจารณาของทางผู้จัดทำมีความเห็นว่าโปรแกรมเกมควรพัฒนาในส่วนการเขียนโปรแกรมคำสั่งบังคับหุ่นยนต์อัตโนมัติต่อไป และเพิ่มเติมการทำงานในส่วนอื่นๆ ได้แก่ ระบบบัญชีผู้เล่นและการเซฟเกม ระบบกระดานคะแนน เพิ่มความหลากหลายของด่านและภารกิจภายในเกม การเคลื่อนที่ของหุ่นยนต์ของฝั่งศัตรู เพิ่มความสวยงามและองค์ประกอบของเกม อาทิเช่น กระจกสุนของหุ่นยนต์ที่ยิงออกมา เสียงภายในเกม เอฟเฟกในการกระทำต่างๆของหุ่นยนต์ หน้า

Interface ภายในเกมและหน้าเมนูหลักให้สวยงามและใช้งานสะดวกขึ้น และระบบการเล่นให้ทำงานได้สมบูรณ์มากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Plarium. 2561. **The Different Between Real-time and Turn-based Strategy.** [Online].

Available :<https://plarium.com/en/blog/rts-vs-tbs>

Google. 2558. **Blockly Game.** [Online].

Available : <https://blockly-games.appspot.com/>

Queen Mary University of London. 2556. **Playing video game can boost brain power.** [Online].

Available :https://www.eurekalert.org/pub_releases/2013-08/qmuo-pvg082113.php

Unity Technologies. 2561. **Unity 3D Tutorial.** [Online].

Available :<https://unity3d.com/learn/tutorials>

Unity Technologies. 2561. **Unity 3D Document.** [Online].

Available :<https://docs.unity3d.com/Manual/index.html>

Yani Widyani. 2557. **Game Development Life Cycle.** [Online].

Available :https://www.researchgate.net/publication/271548605_Game_development_life_cycle_guidelines