



**ข้อเสนอโครงการวิศวกรรมคอมพิวเตอร์**  
**วิชา 01076014 การเตรียมโครงการวิศวกรรมคอมพิวเตอร์**  
**ภาคเรียนที่ 2 ปีการศึกษา 2563**

1. ชื่อหัวข้อโครงการ (ไทย) .....อุปกรณ์สะท้อนแรงเสมือนจริงสำหรับเกมต่อสู้ในโลกความจริงเสมือน.....
2. ชื่อหัวข้อโครงการ (อังกฤษ) ....Force Reflex Arm Control for Action VR Game.....
3. Keyword 3 คำ ..... Force Reflex Arm Control, VR, Reinforcement Learning.....
4. ประเภทโครงการ (✓)
 

✓ 1. HW+SW
☐ 2. SW\_Dev
☐ 3. Research
5. รายชื่อผู้ทำโครงการ
 

5.1. นาย กล้าณรงค์.....ทมโคตร.....รหัส ..... 61010047.....

5.2. นางสาว ศิริวรรณ.....กุลละวณิชย์.....รหัส ..... 61011023.....

5.3. นางสาว สิริินยา.....กำยาน.....รหัส ..... 61011120.....
6. อาจารย์ที่ปรึกษา
 

6.1. อาจารย์ที่ปรึกษาหลัก .....ผศ.ดร.ชมพูนุท จินจาคาม.....

6.2. อาจารย์ที่ปรึกษาร่วม .....อ.สรยุทธ กลมกล่อม.....

## 1. ที่มาและความสำคัญของปัญหา (Motivation)

ปัจจุบันโลกเรานั้นมีสื่อเพื่อมอบความบันเทิงมากมายเพื่อที่จะให้ผู้คนที่หนีออกจากความเครียด หรือโลกความเป็นจริง และดำดิ่งเข้าไปในนั้นไม่ว่าจะเป็นภาพยนตร์ ดนตรี หรือวิดีโอเกม โดยสื่อเหล่านั้นล้วนออกแบบมาเพื่อให้ผู้ใช้ได้ดำดิ่ง และอินไปกับสื่อเหล่านั้น และด้วยเทคโนโลยีในสมัยนี้ทำให้มีเครื่องมือเพื่อเล่นเกมมากมาย และหนึ่งในนั้นก็คือเทคโนโลยี VR หรือ Virtual Reality

เนื่องจากทางกลุ่มผู้จัดทำนั้นก็เป็นนักเล่นเกม และได้ลองเล่นเกม VR ที่ร้านเกม VR จึงทำให้รู้สึกว่าการเล่นเกม VR นั้น ได้ให้ประสบการณ์ที่แปลกใหม่ที่แตกต่างจากที่เคยเล่นเกม ใน PC หรือ Console เพราะว่าเมื่อเราเล่นเกมจะได้เป็น ได้เห็น และได้เคลื่อนไหวเหมือนตัวละครในเกม แต่เมื่อเราได้เล่นเกมที่ใช้อาวุธระยะประชิด เรากลับพบว่าเมื่อได้ลองใช้อาวุธโจมตีดูแล้ว เรากลับรู้สึกเหมือนเราไม่ได้ฟันโดนอะไรเลย ทางเราจึงได้ตัดสินใจที่จะทำเกม และอุปกรณ์ต่อพ่วงเพื่อเติมเต็มความรู้สึกตรงนั้นที่ขาดหายไป

จากที่กล่าวมาข้างต้นทางเราจึงมีความคิดริเริ่มที่จะดำเนินการทำโครงการ Realistic Action Game With Force Reflex VR Gadget ขึ้นเพื่อที่จะได้เติมเต็มความรู้สึก และอรรถรสที่ขาดหายไป โดยทางกลุ่มผู้จัดทำนั้นได้ออกแบบระบบเป็น 3 ส่วนหลัก ๆ คือ Hardware, Software และ AI โดยชิ้นงานหลักของส่วน Hardware คือสร้างตัวอุปกรณ์สะท้อนแรงเสมือนจริง (Force Reflex VR Gadget) เพื่อช่วยให้ผู้เล่นนั้นได้รับประสบการณ์ การฟันโดนในขณะที่เล่นเกม ชิ้นงานหลักของส่วน Software ก็คือสร้างตัวเกม และสภาพแวดล้อม เพื่อให้ผู้เล่นได้รับประสบการณ์ที่ทางผู้สร้างต้องการมอบให้ และชิ้นงานหลักของส่วน AI คือการใช้ Deep Learning เข้ามาทำให้ศัตรูในเกมสามารถต่อสู้กับผู้เล่นได้อย่างเต็มประสิทธิภาพ และสร้างความท้าทายให้กับผู้เล่น

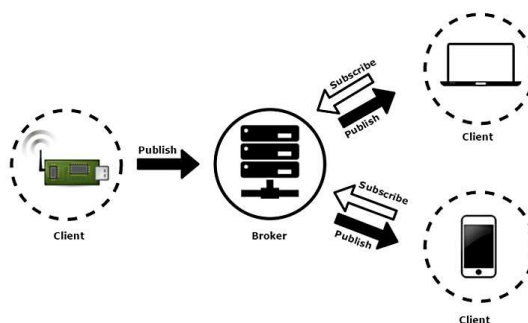
## 2. วัตถุประสงค์ (Objectives)

1. เพื่อสร้างอุปกรณ์สะท้อนแรงเสมือนจริงสำหรับเล่นเกมต่อสู้ในโลก VR
2. เพื่อเพิ่มประสบการณ์และเติมเต็มอรรถรสของผู้เล่นเกม VR (ประเภทเกมต่อสู้ระยะประชิด)
3. เพื่อบูรณาการความรู้ด้าน Hardware, Software และ AI

## 3. ทฤษฎีที่เกี่ยวข้อง (Theoretical Background)

### 3.1 MQTT (MQ TELEMETRY TRANSPORT)

MQTT เป็นโพรโทคอลสื่อสารชั้นแอปพลิเคชันที่รันบน TCP/IP ถูกพัฒนาขึ้นในปี ค.ศ. 1999 โดย IBM และ Eurotech สำหรับการมอนิเตอร์สถานะท่อส่งน้ำมันส่วนที่วางผ่านเขตทะเลทราย ด้วยการออกแบบให้เป็น การรับส่งข้อความที่มีน้ำหนักเบาและเป็นโพรโทคอลเปิด ทำให้ในปัจจุบัน MQTT ถูกนำมาใช้แพร่หลายในการสื่อสารแบบ M2M หรือ IoT เพราะเหมาะสมกับอุปกรณ์ปลายทางที่มีขนาดเล็ก/พลังงานจำกัด หรือในการสื่อสารระยะไกลที่ต้องการการใช้งานแบนด์วิดธ์อย่างมีประสิทธิภาพ



รูป 1 โมเดลการสื่อสารของโปรโตคอล MQTT<sup>1</sup>

**โบรกเกอร์** เป็นจุดศูนย์กลางในการรับส่งข้อความระหว่างไคลเอนต์ วิธีการกำหนดเส้นทาง (Routing) กระทำผ่าน Topic โดยไคลเอนต์ Subscribe ใน Topic ที่ตนต้องการ จากนั้นโบรกเกอร์จะส่งข้อความทั้งหมดที่ถูก Publish ใน Topic นั้นๆ ไปให้ ดังนั้นไคลเอนต์จึงสื่อสารกันได้โดยไม่จำเป็นต้องรู้จักกัน ช่วยลดความเกี่ยวพันระหว่างผู้สร้างข้อมูลและผู้ใช้ข้อมูล

**ไคลเอนต์** จะเป็นได้ทั้ง Publisher หรือ Subscriber หรือ Publisher/Subscriber พร้อมๆ กัน และจะเป็นอุปกรณ์ใดๆ ก็ได้ที่สามารถรัน MQTT Client Library บน TCP/IP Stack การที่ MQTT ใช้โมเดล Publish/Subscribe ตรรกะส่วนใหญ่จึงไปตกอยู่ในฝั่งโบรกเกอร์ ทำให้ Library มีขนาดเล็ก ติดตั้งได้ง่าย ใช้งานได้กับอุปกรณ์ที่มีทรัพยากรจำกัด

เมื่อเปรียบเทียบ MQTT กับ HTTP (REST) ที่มีสถาปัตยกรรมแบบ Request/Response จะพบว่า MQTT มีความได้เปรียบที่โบรกเกอร์สามารถผลัก (Push) ข้อความไปยังไคลเอนต์ได้ตามเหตุการณ์ (Event-driven) ในขณะที่เมื่อใช้ HTTP ฝั่งไคลเอนต์ต้องคอยโพลข้อมูลเป็นระยะๆ และต้องตั้งค่าคาบเวลาการโพลไว้ก่อนล่วงหน้า โดยแต่ละครั้งต้องมีการสร้างการเชื่อมต่อขึ้นใหม่และอาจจะไม่มีข้อมูลใหม่ใดๆ ให้พอเด็ด ดังนั้นหากต้องการให้ระบบทำงานแบบ Real Time หรือใกล้เคียง ย่อมหมายถึงต้องตั้งค่าคาบเวลาการโพลให้สั้น และความสิ้นเปลืองของการใช้ช่องสัญญาณที่ไม่จำเป็นที่ตามมา นี่จึงเป็นอีกเหตุผลสำคัญที่ทำให้ MQTT ได้รับความนิยมเหนือ REST สำหรับการใช้งานแบบ M2M นอกเหนือจากการมีน้ำหนักเบา

**MQTT Topic** เป็น UTF-8 String ในลักษณะเดียวกับ File Path คือสามารถจัดเป็นลำดับชั้นได้ด้วยการขึ้นด้วย “/” ตัวอย่างเช่น myhome/floor-one/room-c/temperature ไคลเอนต์สามารถเลือก Publish หรือ Subscribe เฉพาะ Topic หรือ Subscribe หลาย Topic พร้อมๆ กันโดยใช้ Single-Level Wildcard (+) เช่น myhome/floor-one+/temperature หมายถึงการขอเขียนหรือรับข้อความ temperature จากทุกๆ ห้องของ myhome/floor-one หรือ Multi-Level Wildcard (#) เช่น myhome/floor-one/# หมายถึงการขอเขียนหรือรับข้อความทั้งหมดที่มี Topic ขึ้นต้นด้วย myhome/floor-one เป็นต้น [1]

<sup>1</sup> <https://netpie.gitbooks.io/doc/content/second-chapter.html>

### 3.2 VR (Virtual Reality)

VR (Virtual Reality) เป็นการผสมผสานเทคโนโลยีหลาย ๆ ด้าน ไม่ว่าจะเป็นเทคโนโลยีคอมพิวเตอร์ เทคโนโลยีซีมูเลชัน เทคโนโลยีสารสนเทศ และเทคโนโลยีอื่น ๆ เกิดเป็นเทคโนโลยีที่สามารถสร้างสภาพแวดล้อมเสมือนจริงในรูปแบบ 3 มิติ เป็นการมอบประสบการณ์อันสมจริงให้กับผู้ใช้งาน

จุดประสงค์หลักของเทคโนโลยี VR คือการสร้างสภาพแวดล้อมเสมือนจริงด้วยคอมพิวเตอร์ เพื่อให้ผู้ใช้งานได้รับความรู้สึกสมจริงผ่านทางอุปกรณ์รับชมที่รับสัญญาณมาจากคอมพิวเตอร์ โดยตัวอุปกรณ์รับชมนั้นมีหลายแบรนด์ให้เลือกใช้ เช่น Oculus Rift, HTC Vive เป็นต้น ซึ่งเทคโนโลยีนี้ถือเป็นการพัฒนาวิธีปฏิสัมพันธ์กับผู้ใช้งาน จากที่สื่อหรือเกมแต่ก่อนเป็นการปฏิสัมพันธ์กับผู้ใช้งานผ่านประสาทสัมผัสเดียว (single sensory interaction) แต่เทคโนโลยี VR สามารถทำให้เกิดการปฏิสัมพันธ์กับผู้ใช้งานผ่านหลายประสาทสัมผัส (multi-sensory interaction) ในเวลาเดียวกันได้ เป็นการทำให้ผู้ใช้ดำดิ่งในสื่อและเกมนั้น ๆ มากยิ่งขึ้น [2]

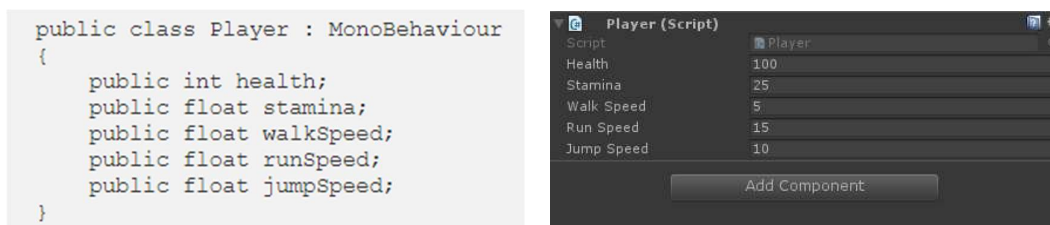


รูป 2 ตัวอย่างภาพที่ผู้ใช้งานมองเห็นเมื่อสวมอุปกรณ์รับชม<sup>2</sup>

### 3.3 Unity

Unity เป็น game engine สำหรับพัฒนาเกมที่ซัพพอร์ตหลากหลายแพลตฟอร์ม เช่น คอมพิวเตอร์ มือถือ เครื่องเกมคอนโซล และ VR เป็นต้น ซึ่ง Unity สามารถสร้างเกมได้หลายรูปแบบ ทั้งแบบ 2 มิติ 3 มิติ เกม VR และ AR และยังสามารถใช้ engine นี้พัฒนาสื่อบันเทิงอื่น ๆ เช่น ภาพยนตร์ ได้อีกด้วย [3]

ปัจจุบัน ภาษาที่ใช้สำหรับพัฒนาเกมใน Unity คือ C# ซึ่งใช้สำหรับสร้างไฟล์ script เพื่อควบคุม object ต่าง ๆ ภายในเกมได้อย่างอิสระ เช่น script หรือก็คือ class ชื่อว่า Player ไว้ใช้ควบคุม object ตัวละครผู้เล่น ซึ่งจะประกอบด้วยตัวแปรต่าง ๆ ที่เกี่ยวข้องกับตัวละครผู้เล่น ดังรูป 3



รูป 3 ตัวอย่าง script ชื่อว่า Player และตัวแปรภายใน<sup>3</sup>

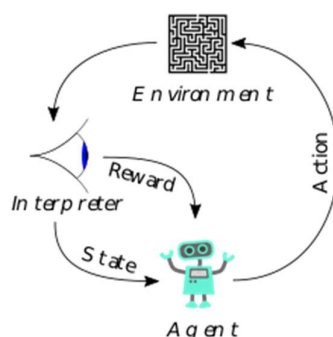
<sup>2</sup> [https://en.wikipedia.org/wiki/Virtual\\_reality](https://en.wikipedia.org/wiki/Virtual_reality)

<sup>3</sup> <https://medium.com/@masads/unity-editor-scripting-a-kick-starter-guide-part-1-faf9532136f0>

นอกจากนี้ ผู้ใช้ Unity สามารถพัฒนา assets เช่น โมเดลตัวละคร แล้วนำมาลงขายหรือแจกจ่ายให้ผู้อื่นผ่าน Unity Asset Store ได้

### 3.4 Reinforcement Learning

Reinforcement Learning จัดเป็นแขนงหนึ่งของ Machine Learning โดยจะเน้นไปในเรื่องของ การสร้างการตัดสินใจ ตัว Agent จะมีการเรียนรู้วิธีการที่จะทำให้บรรลุเป้าหมายในสภาพแวดล้อมที่มีความไม่แน่นอน หรือมีความซับซ้อน โดยจะมีการใช้หลักการ “ลองผิด ลองถูก” เพื่อที่จะได้วิธีที่สามารถใช้ในการแก้ปัญหา และเพื่อที่จะให้ Agent รู้ถึงว่าโปรแกรมเมอร์ต้องการอะไร จะมีการให้รางวัล (reward) หรือการทำให้โทษ (penalty) ให้แก่การกระทำของ Agent โดยเป้าหมายในการทำ reinforcement learning คือ การทำให้ reward มีจำนวน สูงที่สุด [\[4\]](#)



รูป 4 วิธีการเรียนรู้แบบ Reinforcement learning<sup>4</sup>

### 3.5 Imitation Learning

เป็นการเทรน agent โดยใช้การสาธิตให้ agent ดูเป็นตัวอย่างว่าควรจะต้องมีการทำงานอย่างไร ต้องมีลักษณะพฤติกรรมเป็นอย่างไร และยังเป็นการเรียนรู้ในรูปแบบของ “การลองผิดลองถูก” โดย agent จะใช้ตัวอย่างพฤติกรรมที่ได้เรียนรู้จากต้นแบบของมนุษย์ ประกอบกับการลองผิดลองถูก หรือมีการให้รางวัลเมื่อ agent ทำสิ่งที่ดี ในการเทรน agents ด้วยรูปแบบนี้ จะทำให้ agent สามารถเรียนรู้ได้ไวกว่าเดิมมากยิ่งขึ้น

ในการใช้ Imitation learning ยังมีข้อจำกัดอยู่ คือบางที่ agents จะไม่ได้เรียนรู้วิธีใหม่ๆ ที่จะสามารถนำไปสู่ผลลัพธ์ที่ดีขึ้นได้ เพราะ มันอาจจะไม่ได้เรียนรู้วิธีใหม่ๆ มากพอนัก จึงอาจจะต้องมีการนำการเรียนรู้อื่นๆ มาประกอบรวมด้วย เช่น ใช้ reinforcement learning ร่วมกับตัว Imitation learning ด้วย ก็จะทำให้ agents ได้เรียนรู้วิธีการใหม่ๆ ที่จะนำไปสู่การได้ผลลัพธ์ที่ดีขึ้นกว่าเดิมได้นั่นเอง [\[5\]](#)

### 3.6 Unity ML-Agents

เป็นโปรเจกต์ในรูปแบบของ open-source ของทางผู้พัฒนา Unity ที่ใช้ในการสร้างสภาพแวดล้อมสำหรับการเทรนตัว agents โดยจะมีในส่วนของอัลกอริทึมที่เป็นที่นิยม และเป็นที่น่าสนใจเตรียมไว้ให้ เพื่อที่จะช่วยให้ผู้พัฒนาเกมหรือผู้ที่พัฒนาเกมเป็นงานอดิเรก ง่ายต่อการที่จะเทรน AI ในเกม 2D, 3D และ VR/AR และสำหรับผู้

<sup>4</sup> Reinforcement Learning - Wikipedia

ที่ทำวิจัย ก็จะมีการเตรียมส่วนที่ใช้ในการเชื่อมต่อตัว API ของ Python เพื่อที่จะเทรน agents โดยใช้ Reinforcement learning, Imitation learning, neuroevolution หรือวิธีอื่นๆ โดยในการเทรน agents สามารถนำไปใช้ได้หลายวัตถุประสงค์ เช่น ใช้ในการควบคุมพฤติกรรมของ NPC [5]

#### 4. งานวิจัยที่เกี่ยวข้อง (Related Works)

##### 4.1 How Fast is MQTT? Statistical Model Checking and Testing of IoT Protocols

งานวิจัยเกี่ยวกับการใช้โมเดลทางสถิติเข้ามาช่วยทำนายพฤติกรรมของโปรโตคอล MQTT โดยมี Mosquitto เป็น Broker จากนั้นนำ Log เพื่อมาตรวจสอบ และทำ Linear Regression เพื่อหาการแจกแจงของ Latency จากนั้นก็นำมาทดสอบเพื่อดูพฤติกรรมของ Client แล้วนำมาสร้างเป็นโมเดลจำลองเพื่อคำนวณว่าเมื่อมี Client มากขึ้นจะมี Latency ประมาณเท่าใด

จากการวิจัย สรุปได้ว่า MQTT มีเวลาเฉลี่ย จากโมเดลจำลอง โดยใช้ส่งข้อมูลจำลองจำนวน 1,060 ข้อมูล ตัวอย่างพบว่าสำหรับ Clients จำนวน 50 70 90 110 130 และ 150 clients นั้นใช้เวลา 4:28 4:49 4:57 5:05 5:15 และ 5:23 นาทีตามลำดับโดยเวลานั้นรวมเวลาในการ Connect Publish Subscribe และ Unsubscribe ของแต่ละ Clients [6]

##### 4.2 Design Patterns Applied for Networked First Person Shooting Game Programming

งานวิจัยเกี่ยวกับการศึกษา design patterns รูปแบบต่าง ๆ เพื่อนำมาประยุกต์ใช้กับการพัฒนาเกมแนว FPS (First Person Shooting) ซึ่งการออกแบบเกมโดยใช้แนวคิดของ design patterns จะช่วยเพิ่มคุณภาพให้กับตัวซอฟต์แวร์เกม คือ ช่วยให้สะดวกต่อการนำมาใช้ซ้ำ (reusability) ง่ายต่อการแก้ไขและพัฒนา (maintainability) และเพื่อให้ขยายความสามารถของระบบได้ (extensibility) โดยในงานวิจัยนี้ได้ศึกษาและคัดเลือก design patterns ที่เหมาะสมกับส่วนต่าง ๆ ของเกม ได้แก่ การจัดการ states ภายในเกม, อาวุธภายในเกม, ระบบออนไลน์, การเรนเดอร์ศัตรูและแมพภายในเกม [7]

จากการวิจัย สรุปได้ว่ามี design patterns ที่แนะนำ ดังนี้

- Builder pattern เหมาะสำหรับการออกแบบระบบแต่งตัว (customize) ของตัวละครผู้เล่น
- State pattern เหมาะสำหรับการออกแบบ states ของเกม
- Factory method pattern และ Decorator pattern เหมาะสำหรับการออกแบบระบบอาวุธ
- Prototype pattern เหมาะสำหรับการออกแบบการเรนเดอร์ศัตรู
- Observer pattern เหมาะสำหรับการออกแบบระบบการเล่นแบบออนไลน์

##### 4.3 Applying Design Patterns in Game Programming

งานวิจัยเกี่ยวกับการศึกษา design patterns เพื่อนำมาประยุกต์ใช้กับการเขียนเกมโดยใช้ภาษา C# ด้วยซอฟต์แวร์ XNA โดยศึกษาและคัดเลือก design patterns ที่เหมาะสมกับส่วนต่าง ๆ ของเกม ได้แก่ การจัดการ sprites, การจัดการ states ภายในเกม และการจัดการการชน (collision) ระหว่าง object ภายในเกม [8]

จากการวิจัย สรุปได้ว่ามี design patterns ที่แนะนำ ดังนี้

- Strategy pattern และ Command pattern เหมาะสำหรับการออกแบบส่วนของ sprites
- State pattern เหมาะสำหรับการออกแบบ states ของเกม
- Observer pattern หรือ Mediator pattern เหมาะสำหรับการออกแบบ objects ภายในเกม
- Visitor pattern เหมาะสำหรับการออกแบบส่วนของการชน (collision) ระหว่าง objects

#### 4.4 Creating Pro-Level AI for a Real-Time Fighting Game Using Deep Reinforcement Learning

งานวิจัยเกี่ยวกับการใช้ Deep reinforcement learning ในการทำคู่ต่อสู้ในเกม Blade & Soul โหมด PvP (Player versus Player) โดยมีเป้าหมายให้คู่ต่อสู้ที่มีความสามารถที่เหนือกว่ามนุษย์ โดยมีการใช้เทคนิค self-play curriculum และ data skipping อีกทั้งยังมีการสร้าง Agent ที่มีรูปแบบการต่อสู้ที่แตกต่างกัน 3 รูปแบบ ได้แก่ Aggressive mode, Balance mode, defensive mode เพื่อให้ Agent ได้เรียนรู้รูปแบบการต่อสู้ที่หลากหลาย และสามารถรับมือกับการต่อสู้ได้หลายรูปแบบ

จากงานวิจัย ทีมวิจัยได้ลองนำคู่ต่อสู้ (Agent) ที่ถูกเทรนโดยอัลกอริทึมที่พวกเขาได้คิดค้น และทดลองมาทดสอบกับ Pro-gamers ที่มีชื่อเสียงโด่งดังที่เคยเป็นแชมป์ในการแข่งขัน B&S world Championship โดยผลที่ได้ออกเป็นไปดังรูปที่ 5 โดยจะสังเกตได้ว่า Agent ที่เป็นโหมด Aggressive จะมีเปอร์เซ็นต์ของการชนะที่สูงเป็นอันดับที่ 1 รองลงมาเป็นโหมด Defensive และลำดับสุดท้ายเป็นโหมด Balanced โดยที่โหมด Aggressive มีอัตราการชนะที่สูงกว่าโหมดอื่นๆ อาจจะเป็นเพราะว่าโหมด Aggressive มีการโจมตีอยู่ตลอดเวลา โดยไม่เว้นช่วงเวลาให้มนุษย์ได้หยุดโจมตี แต่ในความเป็นจริงแล้วมนุษย์ไม่สามารถที่จะมีการโจมตีที่ต่อเนื่องได้เหมือนกับ AI ซึ่งตรงนี้อาจจะเป็นข้อเสียเปรียบระหว่างมนุษย์กับ AI จึงทำให้ในโหมด Aggressive นั้นมีเปอร์เซ็นต์ของการชนะที่สูงกว่าโหมดอื่นๆ นั่นเอง [9]

	Aggressive	Balanced	Defensive
Pro-Gamer 1	5-1	2-1	1-2
Pro-Gamer 2	4-0	2-4	4-1
Blind Match	2-0	1-2	0-2
Total	11-1 (92%)	5-7 (42%)	5-5 (50%)

รูป 5 ตารางแสดงเปอร์เซ็นต์การชนะของ AI ที่ต่อสู้กับ Pro-gamers<sup>5</sup>

#### 5. ขอบเขตของโครงการ (Scope)

1. อุปกรณ์สะท้อนแรงเสมือนจริงนั้นทำให้ใช้งานร่วมกันกับเกมที่ทำมาเฉพาะเท่านั้น
2. อุปกรณ์สะท้อนแรงเสมือนจริงนั้นทำให้ผู้เล่นรู้สึกถึงแรงในส่วนของมือ และแขนเท่านั้น
3. ถังมือของอุปกรณ์สะท้อนแรงเสมือนจริงนั้นทำให้ผู้เล่นรู้สึกถึงแรงในการจับอาวุธเท่านั้น

<sup>5</sup> (1) (PDF) Creating Pro-Level AI for Real-Time Fighting Game with Deep Reinforcement Learning (researchgate.net)

4. ส่วนที่ควบคุมแขนของอุปกรณ์สะท้อนแรงเสมือนจริงนั้นทำให้ผู้เล่นรู้สึกถึงแรงในการโจมตีโดน และการป้องกันการโดนโจมตีเท่านั้น
5. เกมที่พัฒนาขึ้นมานั้นมีพื้นที่ที่จำกัด คือผู้เล่นสามารถเคลื่อนไหวได้อิสระภายในพื้นที่ภายในเกมที่เรียกว่าสนามประลอง และห้องเตรียมตัวเท่านั้น
6. เกมที่พัฒนาขึ้นมานั้นสามารถใช้งานร่วมกับอุปกรณ์สะท้อนแรงเสมือนจริงที่พัฒนาขึ้นมาเองเท่านั้น
7. เกมที่พัฒนาขึ้นมานั้นสามารถเล่นแบบออฟไลน์เท่านั้น
8. เกมที่พัฒนาขึ้นมานั้นรองรับผู้เล่นเพียง 1 คน (single-player) เท่านั้น
9. ปัญหาประติษฐ์ของศัตรูโจมตีโดยใช้รูปแบบของท่าที่มีเท่านั้น
10. ปัญหาประติษฐ์ของศัตรูป้องกันโดยใช้รูปแบบของท่าที่มีเท่านั้น
11. ปัญหาประติษฐ์ของศัตรูจะมีการเคลื่อนที่ได้เพียงการเดินเท่านั้น

## 6. การพัฒนาโครงการ (Project Development)

### 6.1 ขั้นตอนการพัฒนา (Methodology)

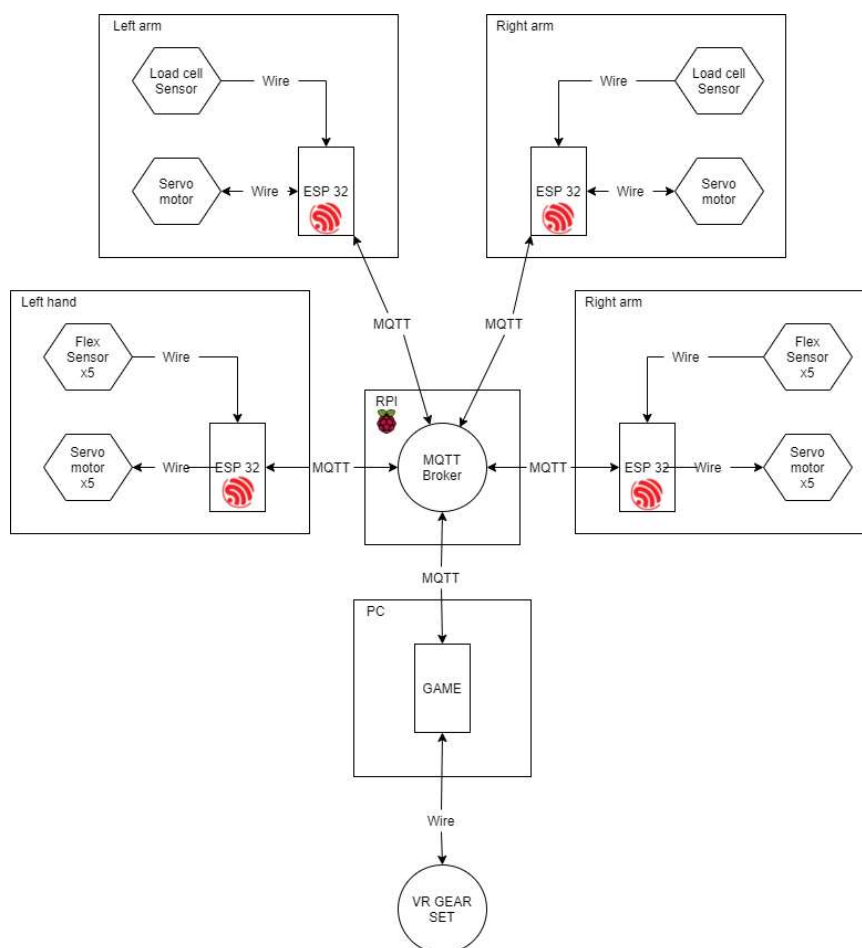
1. กำหนดขอบเขต เป้าหมาย และวัตถุประสงค์ในการทำโครงการ
2. กำหนดหน้าที่และส่วนที่ต้องรับผิดชอบของแต่ละคน
3. ค้นหาข้อมูลตามส่วนที่รับผิดชอบ
4. ศึกษาและทดสอบตามข้อมูลที่ได้มา
5. ออกแบบและปรับใช้กับเป้าหมายของโครงการ
6. ศึกษาและทดสอบตามที่ออกแบบมา
7. นำงานของทุกส่วนมารวมกัน
8. ทดสอบการทำงานร่วมกัน และแก้ไข

### 6.2 การออกแบบ (Design)

#### การออกแบบส่วนของฮาร์ดแวร์

จะมี Raspberry Pi ทำหน้าที่เป็น MQTT Broker รับและส่งข้อมูลให้กับ มือซ้าย/ขวา แขนซ้าย/ขวา และ PC โดยฝั่งที่เป็นมือจะรับข้อมูลที่จะบังคับเซอร์โวแต่ละนิ้วว่าจะควบคุมหรือไม่อย่างไรจาก Broker และจะส่งค่าที่ได้จาก Flex sensor ออกไปยัง Broker ฝั่งที่เป็นแขนจะรับ Servo state จาก Broker และส่งค่าจาก Load cell ไปยัง Broker ฝั่งของ PC จะรับค่านิ้วแต่ละนิ้ว และ Load cell ของทั้งสองข้างจาก Broker รับค่าจาก sensor ของ VR gear set เพื่อนำมาใช้กับเกม และส่งค่านิ้วมือว่าต้องปรับ หรือไม่ และ Servo state ไปยัง Broker



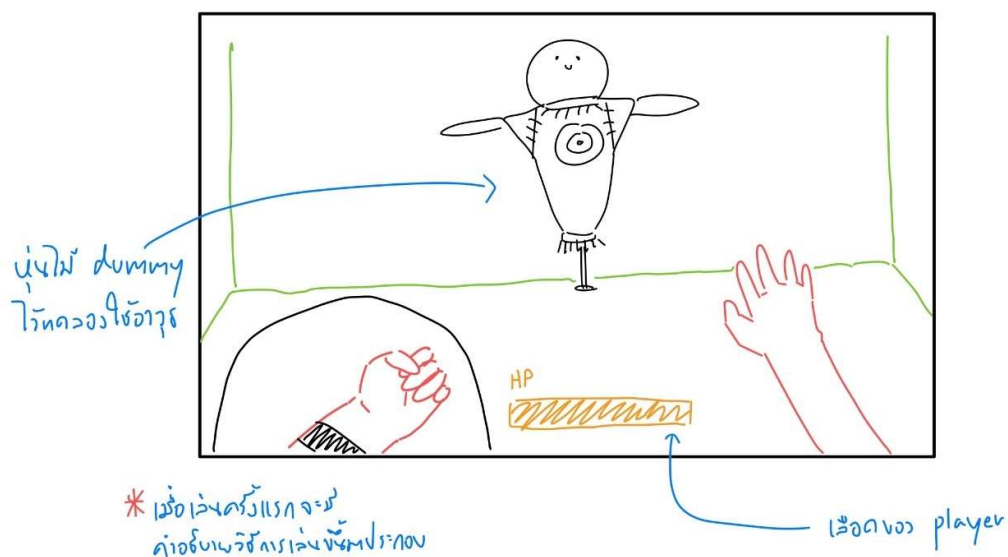


รูป 6 System diagram

## การออกแบบส่วนของเกม

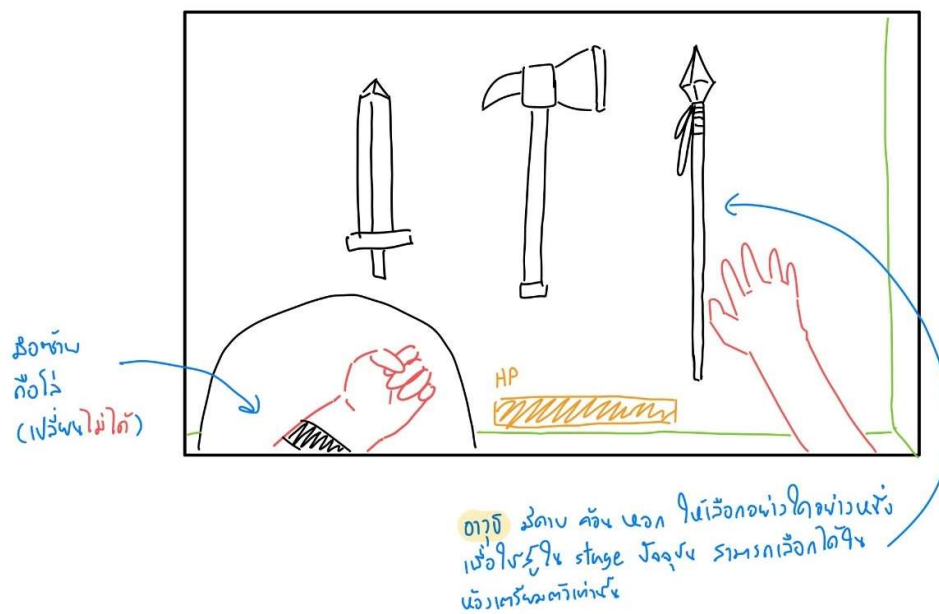
## 1. Storyboard

#0 ห้องเตรียมตัว (ผู้เล่นทดลองใช้อาวุธและทดลองเคลื่อนที่)



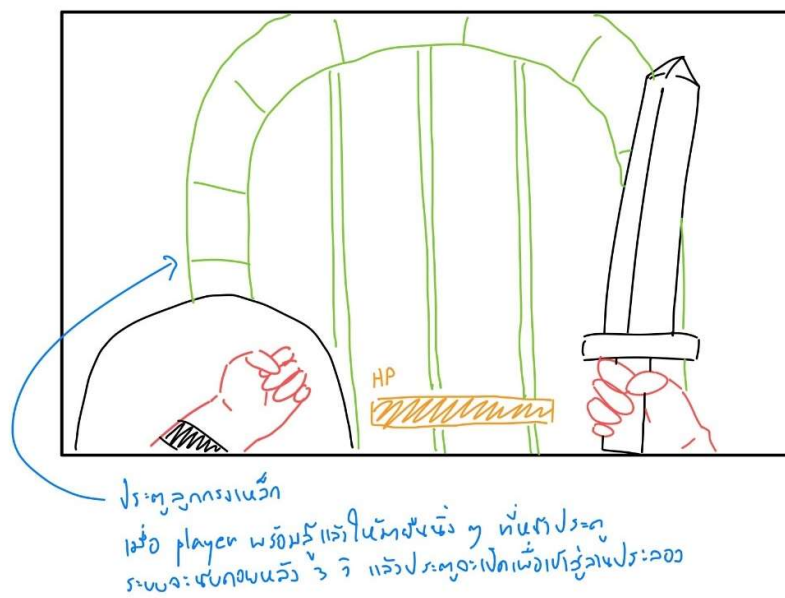
รูป 7 Storyboard scene 0

#1 ห้องเตรียมตัว (ระบบจะนำผู้เล่นเข้ามาที่ห้องนี้เพื่อเตรียมตัวต่อสู้ในลานประลองถัดไป)



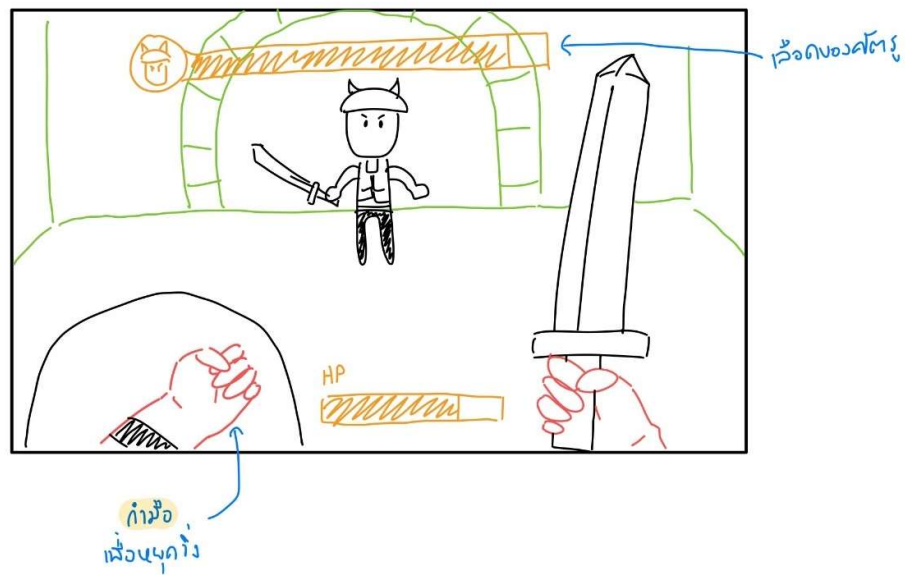
រូប 8 Storyboard scene 1

#2 ห้องเตรียมตัว (ผู้เล่นยืนหน้าประตูเมื่อพร้อม)



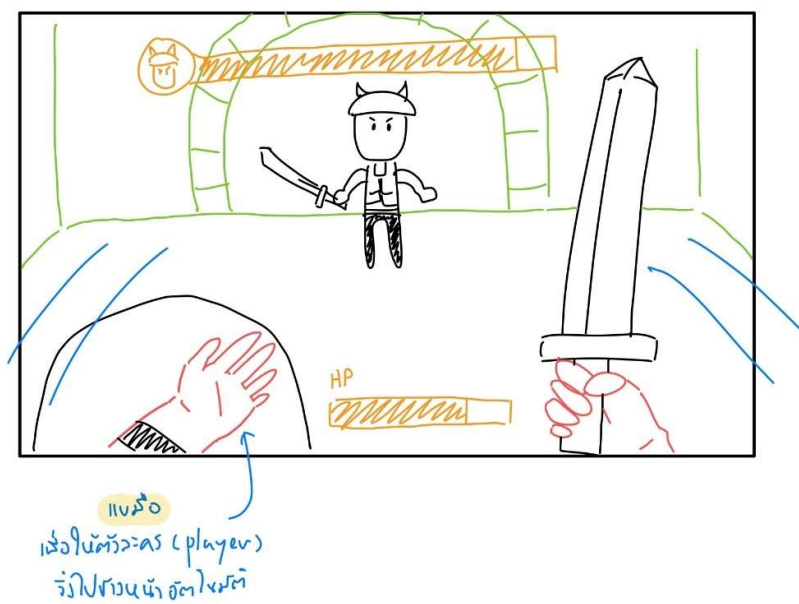
រូប 9 Storyboard scene 2

#3 ลานประลอง (ผู้เล่นต่อสู้กับศัตรู ผู้เล่นสามารถกำมือซ้ายเพื่อหยุดวิ่ง)



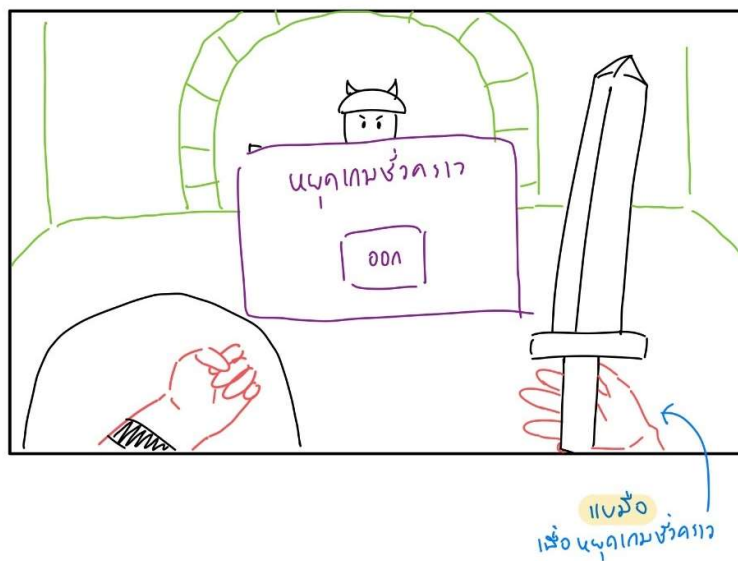
รูป 10 Storyboard scene 3

#4 ลานประลอง (ผู้เล่นต่อสู้กับศัตรู ผู้เล่นสามารถแบมือซ้ายเพื่อวิ่ง)



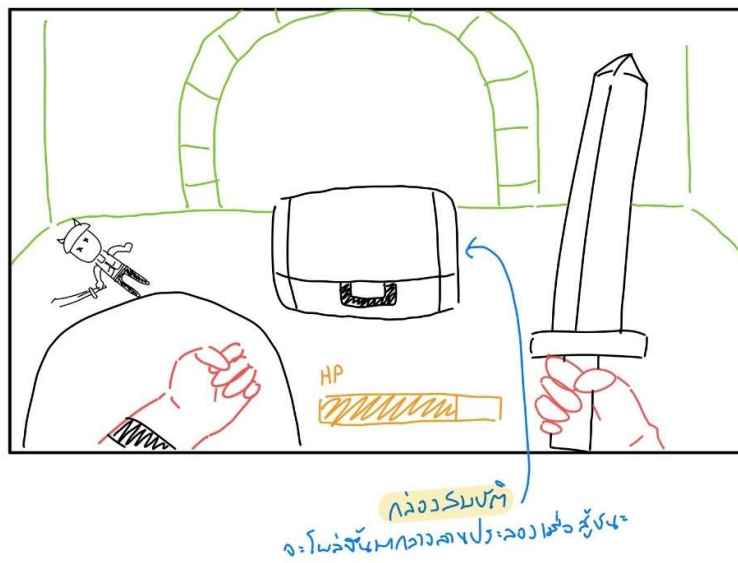
รูป 11 Storyboard scene 4

#5 ลานประลอง (ผู้เล่นแบมือขวาเพื่อหยุดเกมชั่วคราว)



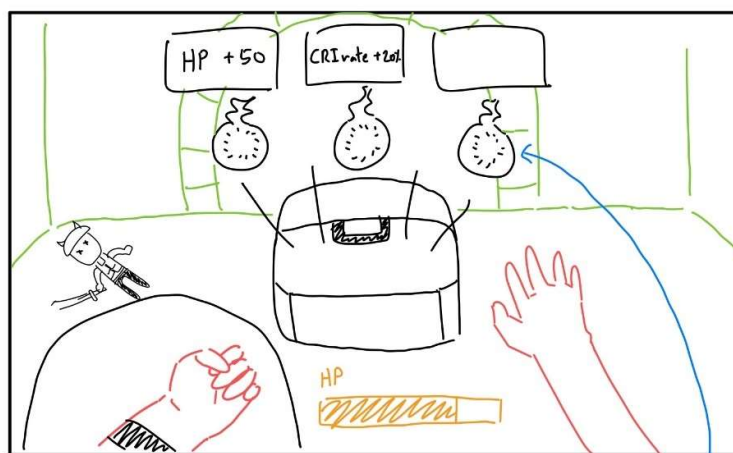
รูป 12 Storyboard scene 5

#6 ลานประลอง (ผู้เล่นชนะศัตรู กล้องสมบัติปรากฏ)



รูป 13 Storyboard scene 6

## #7 ลานประลอง (ผู้เล่นเปิดกล่องสมบัติ ลูกแก้วช่วยเหลือปรากฏ)



Onlos ลูกแก้วรับ status ของผู้เล่น  
โดยเมื่อได้ของใดของหนึ่งจาก 3 ชิ้น  
(ซึ่งพวก stage ต่อไป)

รูป 14 Storyboard scene 7

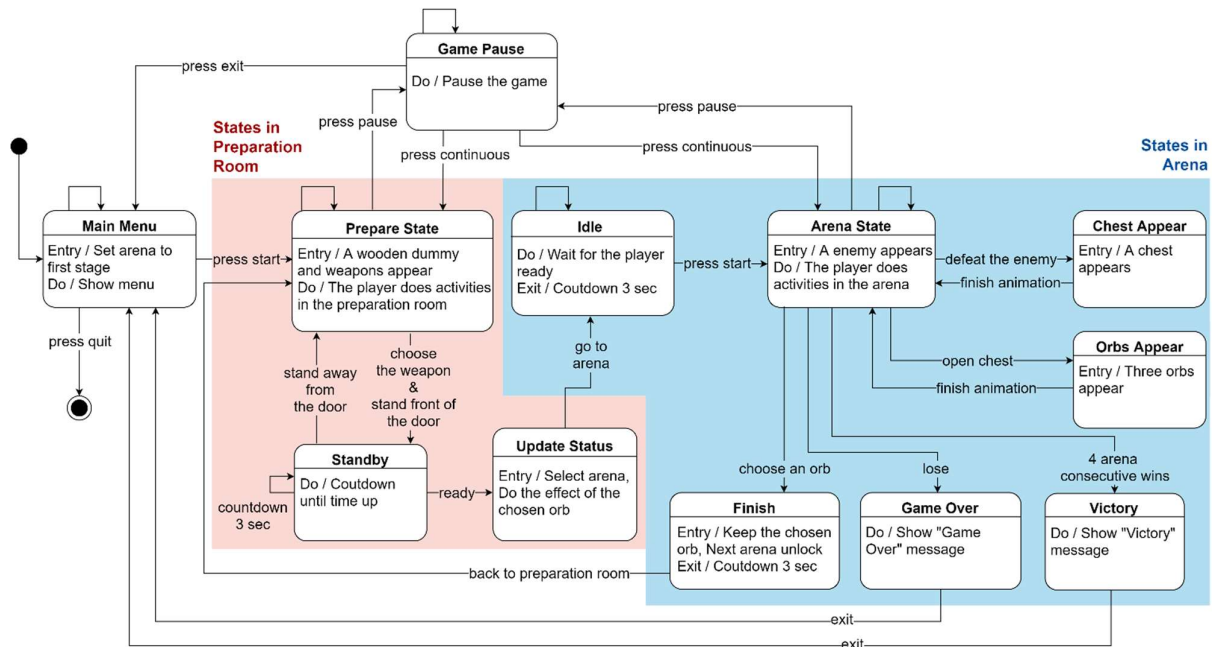
## 2. Requirement

## ตารางที่ 1 Requirement สำหรับเกม

ID	Details	Type	Priority
1	เกมสามารถส่งและรับข้อมูลกับบอร์ด Raspberry Pi ได้	Products Functional	MustHave
2	เกมสามารถเชื่อมต่อกับบอร์ด Raspberry Pi ผ่านทาง wifi ได้	Products Functional	MustHave
3	เกมสามารถเชื่อมต่อกับแว่น VR ได้	Products Functional	MustHave
4	เกมสามารถเชื่อมต่อกับชุดตรวจจับอุปกรณ์สะท้อนแสงเสมือนจริง	Products Functional	MustHave
5	เกมสามารถทำงานร่วมกับระบบ AI ที่พัฒนาขึ้นเองได้	Products Functional	MustHave
6	เกมสามารถพัฒนาบน Unity ได้	Products Functional	MustHave
7	เกมสามารถแสดงผลเป็นภาพ 3 มิติได้	Products Functional	MustHave
8	เกมสามารถพัฒนาด้านขั้นต่ำ 4 ด้านได้	Products Functional	MustHave
9	ในด้านที่ 1 ศัตรูที่ปรากฏมีโอกาสใช้ดาบหรือหอก	Products Functional	MustHave
10	ในด้านที่ 2 ศัตรูที่ปรากฏจะถือโล่ และมีโอกาสใช้ดาบ ค้อน หรือหอก	Products Functional	MustHave
11	ในด้านที่ 3 ศัตรูที่ปรากฏจะถือโล่ และมีโอกาสใช้ดาบ ค้อน หรือหอก	Products Functional	MustHave
12	ในด้านที่ 4 ศัตรูที่ปรากฏจะถือโล่ ใส่เกราะบางส่วน และมีโอกาสใช้ดาบ ค้อน หรือหอก	Products Functional	MustHave
13	เกมสามารถเพิ่มปริมาณเลือดของศัตรูให้มากขึ้นตามด่านที่สูงขึ้นได้	Products Functional	MustHave
14	เกมสามารถเพิ่มความเร็วของศัตรูให้มากขึ้นตามด่านที่สูงขึ้นได้	Products Functional	WantToHave
15	เกมมีห้องเตรียมตัวเพื่อให้ผู้เล่นฝึกเคลื่อนไหวและใช้อาวุธก่อนเริ่มเล่นจริง	Products Functional	MustHave
16	เกมสามารถแสดงผลวัตถุ คือ หุ่นไม้ และอาวุธทั้ง 3 อย่าง ภายในห้องเตรียมตัวได้	Products Functional	MustHave

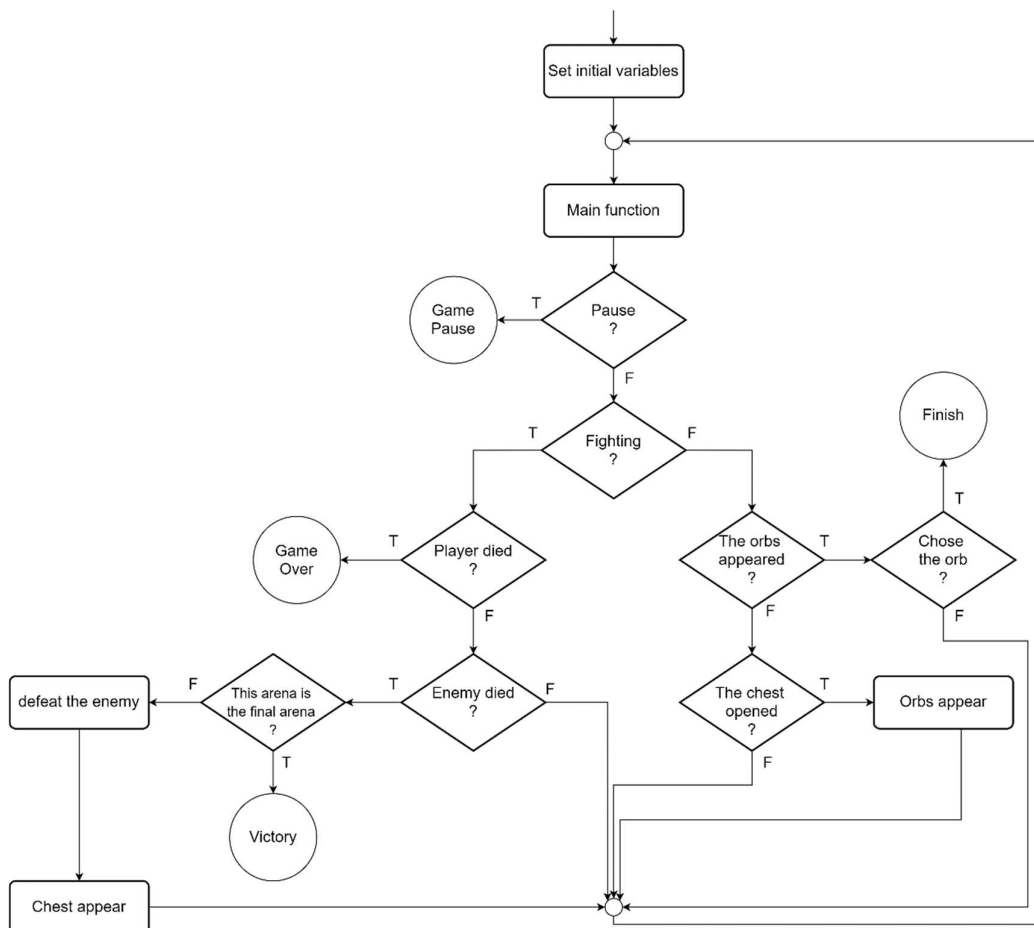
ID	Details	Type	Priority
17	เกมสามารถแสดงผลวัตถุ คือ ศัตรู อาวุธของศัตรู ภายในแต่ละด่านได้	Products Functional	MustHave
18	เกมสามารถแสดงผลสภาพแวดล้อมของแต่ละด่านในลักษณะสนามประลองได้	Products Functional	MustHave
19	เกมสามารถแสดงผลเลือดของผู้เล่นและศัตรูได้	Products Functional	ShouldHave
20	เกมสามารถแสดงผลตัวเลขความเสียหายตามอาวุธที่กระทำได้	Products Functional	ShouldHave
21	เกมสามารถแสดงผลกล่องสมบัติเมื่อจบด่านนั้น ๆ	Products Functional	MustHave
22	เกมสามารถแสดงผลลูกแก้วช่วยเหลือ 3 ลูก และข้อความรายละเอียดของแต่ละลูก เมื่อผู้เล่นเปิดกล่องสมบัติ	Products Functional	MustHave
23	เกมสามารถส่งลูกแก้วช่วยเหลือที่จะแสดงผลภายในด่านนั้น ๆ	Products Functional	MustHave
24	เมื่อจบด่าน เกมสามารถนำผู้เล่นออกจากสนามประลองและพากลับไปยังห้องเตรียมตัวได้	Products Functional	MustHave
25	เมื่อผู้เล่นต้องการออกจากห้องเตรียมตัว เกมสามารถนำผู้เล่นออกจากห้องเตรียมตัวและพาไปยังสนามประลองได้	Products Functional	MustHave
26	เกมสามารถคงปริมาณเลือดของผู้เล่นให้เท่าเดิมเมื่อเปลี่ยนด่าน	Products Functional	MustHave
27	เกมสามารถส่งผลตามลูกแก้วช่วยเหลือที่ผู้เล่นเลือกได้อย่างถูกต้อง	Products Functional	MustHave
28	ภาพกราฟิกของเกมมีความสวยงามสมจริง	Products Functional	WantToHave
29	เกมอนุญาตให้ผู้เล่นสามารถเลือกตาม ด่าน หรือหอก เป็นอาวุธได้	User Interface Functional	MustHave
30	เกมสามารถให้ผู้เล่นใช้มือซ้ายถือโลเพื่อป้องกันความเสียหายที่ได้รับ	User Interface Functional	MustHave
31	เกมสามารถให้ผู้เล่นใช้มือขวาก่ออาวุธโจมตีศัตรูเพื่อสร้างความเสียหายได้	User Interface Functional	MustHave
32	เกมสามารถให้ผู้เล่นกำมือซ้ายเพื่อหยุดการเคลื่อนไหวของตัวเอง	User Interface Functional	MustHave
33	เกมสามารถให้ผู้เล่นแบมือซ้ายเพื่อเคลื่อนไหวตัวละครไปข้างหน้าอัตโนมัติ	User Interface Functional	MustHave
34	เกมสามารถให้ผู้เล่นแบมือขวาเพื่อหยุดเกมชั่วคราวได้	User Interface Functional	ShouldHave
35	เกมสามารถให้ผู้เล่นกำมือขวาเพื่อเล่นต่อ	User Interface Functional	MustHave
36	เมื่อกล่องสมบัติปรากฏขึ้นแล้ว เกมสามารถให้ผู้เล่นปล่อยอาวุธในมือข้างขวา เพื่อเปิดกล่องสมบัติได้	User Interface Functional	MustHave
37	เมื่อผู้เล่นเปิดกล่องสมบัติแล้ว เกมสามารถให้ผู้เล่นใช้มือข้างขวายืนลูกแก้วช่วยเหลือได้	User Interface Functional	MustHave
38	เกมอนุญาตให้ผู้เล่นสามารถเลือกลูกแก้วช่วยเหลือ 1 ลูกจากทั้งหมด 3 ลูกภายในด่านนั้น ๆ	User Interface Functional	MustHave
39	เกมสามารถให้ผู้เล่นชนะได้เมื่อผ่าน 4 ด่านติดต่อกัน	User Interface Functional	MustHave
40	เกมสามารถให้ผู้เล่นผ่านด่านได้เมื่อชนะศัตรู	User Interface Functional	MustHave
41	เกมสามารถให้ผู้เล่นชนะศัตรูได้เมื่อเลือดของศัตรูเท่ากับ 0	User Interface Functional	MustHave
42	เกมสามารถให้ผู้เล่นแพ้ศัตรูได้เมื่อเลือดของผู้เล่นเท่ากับ 0	User Interface Functional	MustHave
43	เกมสามารถให้ผู้เล่นเกมโอเวอร์และต้องเริ่มเกมใหม่ทันทีเมื่อแพ้ศัตรู	User Interface Functional	MustHave

### 3. Finite State Machine ของระบบเกมทั้งหมด



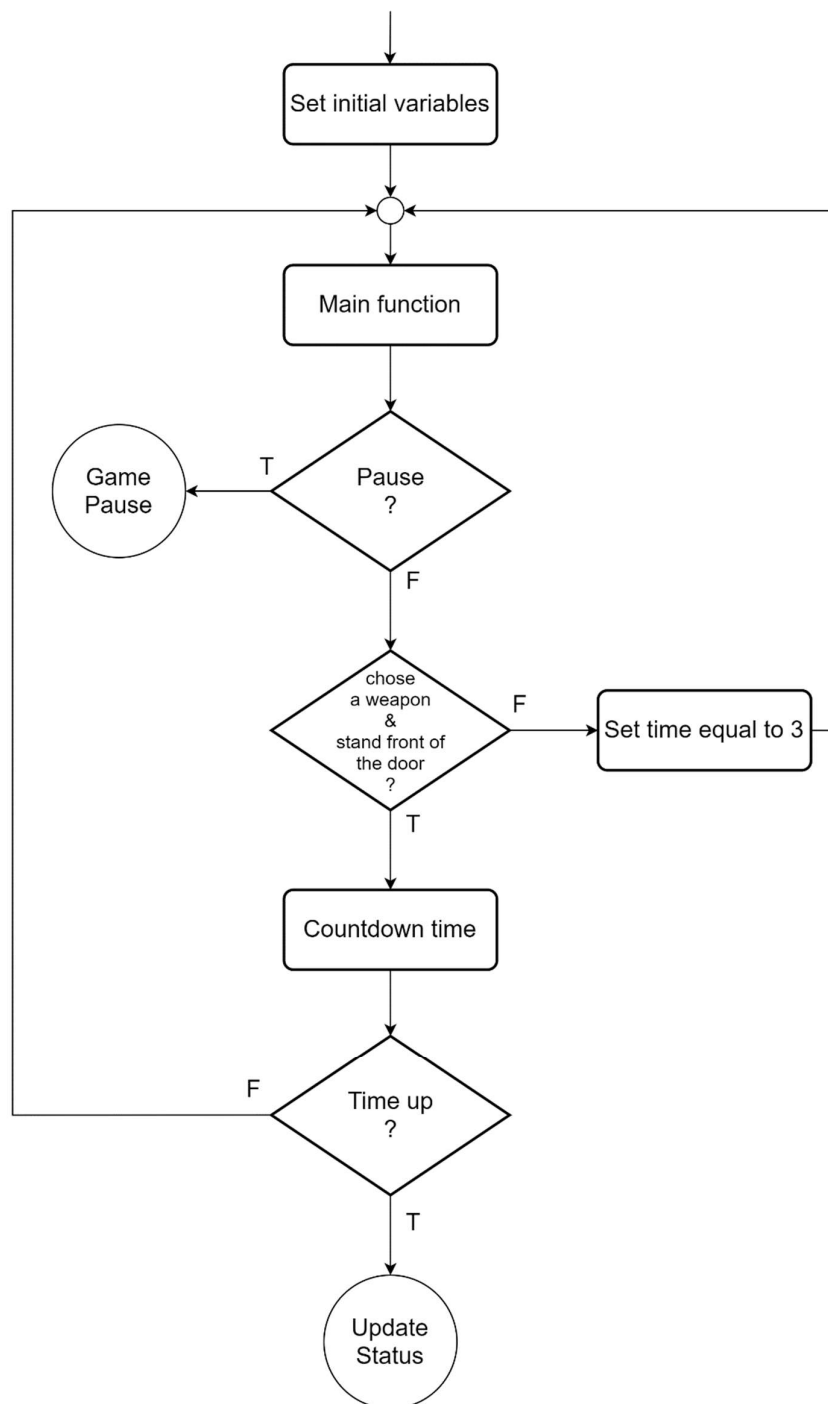
รูป 15 Finite State Machine

### 4. Flowchart (in Arena)



รูป 16 Flowchart (in Arena)

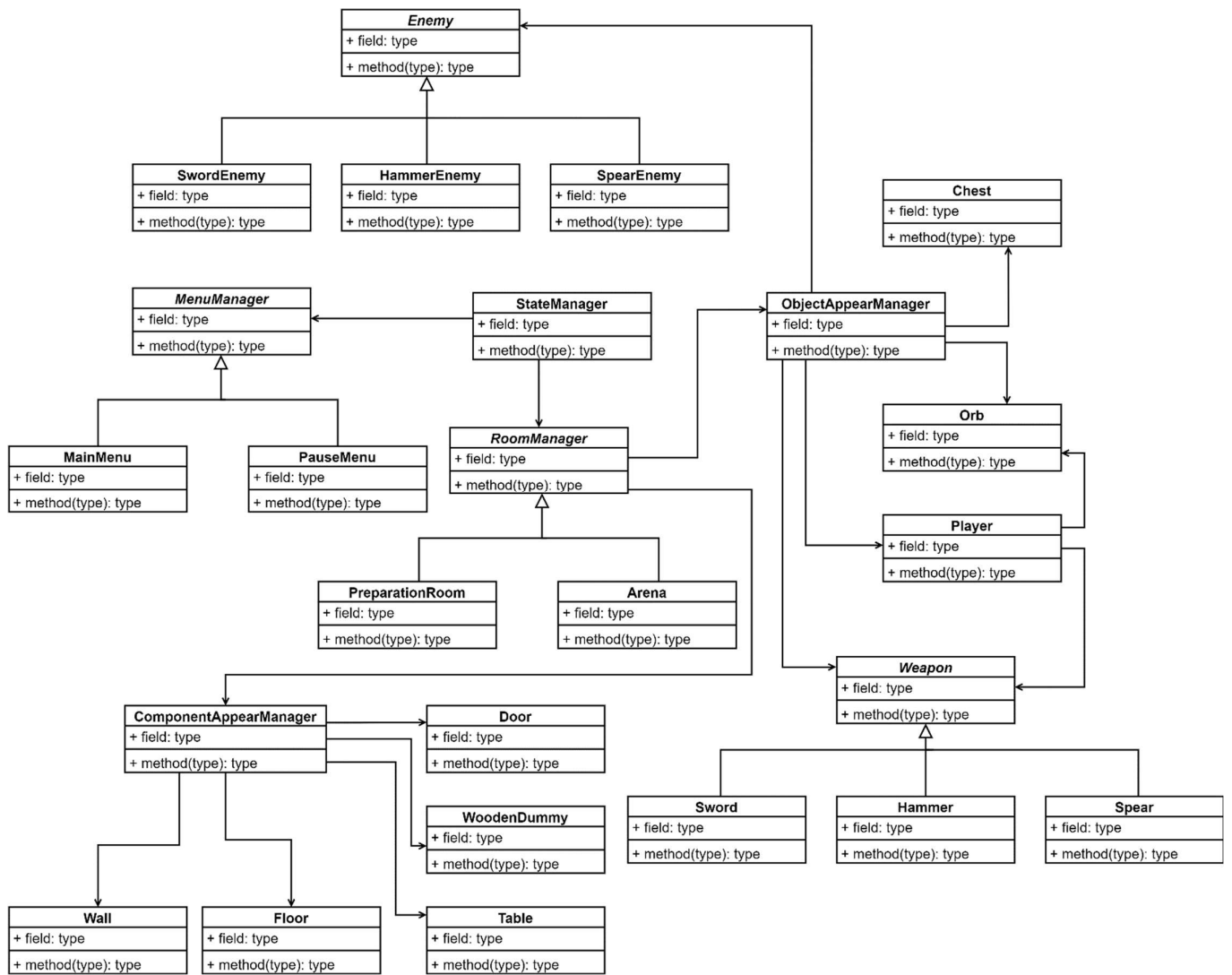
## 5. Flow chart (in Preparation Room)



รูป 17 Flow chart (in Preparation Room)

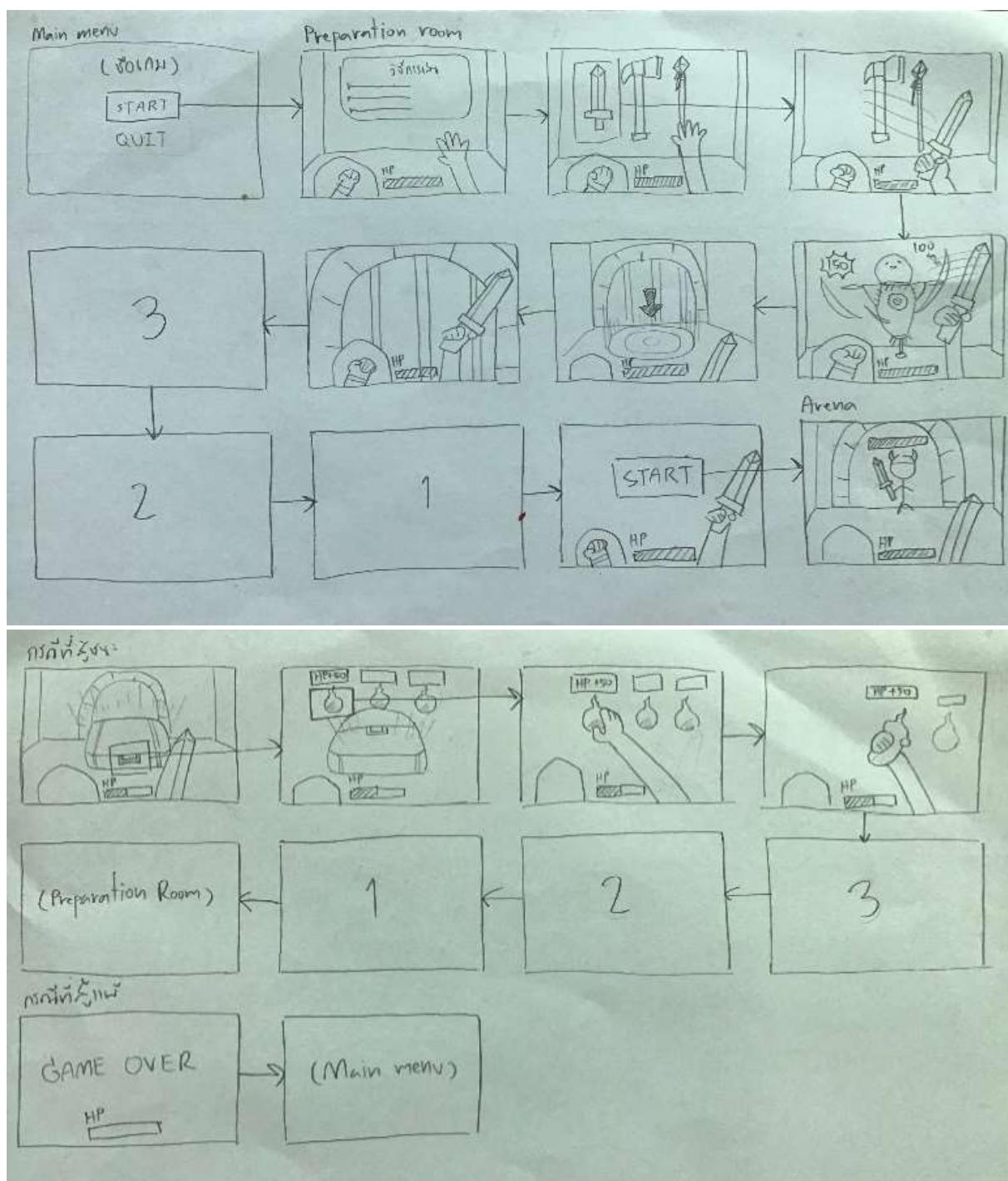


## 6. Class Diagram



รูป 18 Class Diagram

## 7. Wireframe



รูป 19 Wireframe

## การออกแบบปัญญาประดิษฐ์ภายในเกม

### 1. การเทรนศัตรูในเกม

ศัตรูจะมีอยู่ 5 ประเภท ได้แก่

- ศัตรูที่ถือดาบ
- ศัตรูที่ถือหอก
- ศัตรูที่ถือดาบโล่
- ศัตรูที่ถือหอกโล่
- ศัตรูที่ถือค้อนโล่

ผู้เล่นจะมีอยู่ 3 ประเภท ได้แก่

- ผู้เล่นที่ถือดาบโล่
- ผู้เล่นที่ถือหอกโล่
- ผู้เล่นที่ถือค้อนโล่

ซึ่งในการเทรนตัวศัตรูนั้น เราจะนำรูปแบบทั้งหมดที่เป็นได้ของผู้เล่นมาใช้ในการเทรนตัวศัตรูให้สามารถรับมือกับผู้เล่นที่ใช้อาวุธในแต่ละรูปแบบได้ เพื่อเพิ่มความท้าทายให้แก่ผู้เล่น

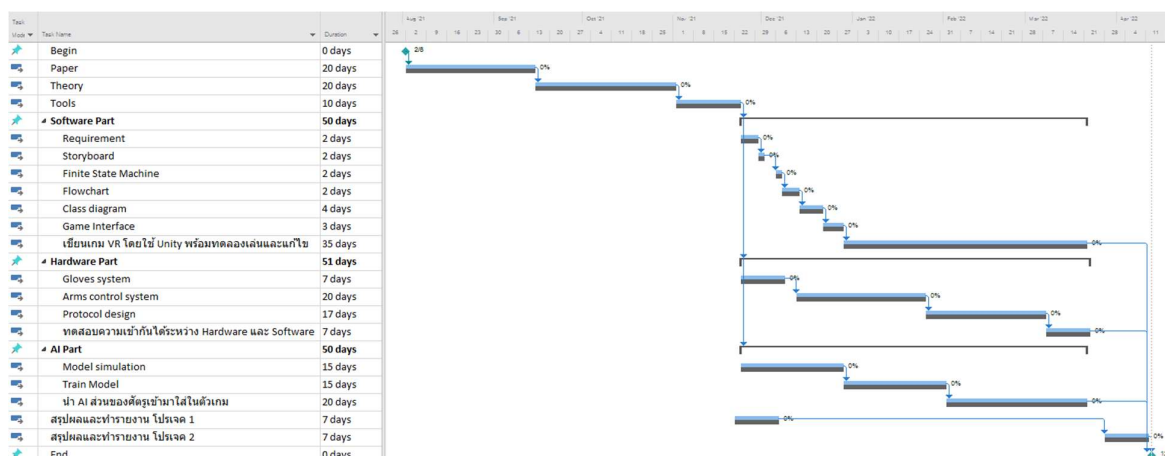
### 2. รูปแบบการเดินของศัตรู

รูปแบบการเคลื่อนที่ของศัตรูในตัวเกมจะปรับให้มีการตัดสินใจเป็นช่วงของ Time step เช่น ถ้าศัตรูตัดสินใจที่จะไปทางซ้าย ก็ให้มีการเคลื่อนที่ไปทางซ้ายด้วยระยะเวลา 10 Time step เป็นต้น จะไม่ให้มีการตัดสินใจในการเคลื่อนที่ของศัตรูในทุก Time step เพราะว่าจะส่งผลให้ศัตรูมีรูปแบบการเดินที่ไม่เป็นธรรมชาติ และมีรูปแบบการเดินที่ส่ายไปส่ายมา

## 6.3 แนวทางการทดสอบและการวัดประสิทธิภาพ (Test and Performance Evaluation Approaches )

1. เช็คว่าสามารถใช้งาน step motor แทน servo motor ได้หรือไม่
2. เช็คการตอบสนองในขณะที่ใช้งานหลายๆ node ว่าการตอบสนองสอดคล้องกับเกมหรือไม่
3. เช็คตัวอัลกอริทึมที่ใช้ว่าทำงานได้ตามที่ต้องการหรือไม่
4. ประเมินความสามารถของตัว AI ที่ได้ทำการเทรนว่ามีลักษณะการเดิน การต่อสู้เป็นไปตามที่ต้องการหรือไม่ มีความยากง่ายมากน้อยเพียงใด โดยจัดทำเป็นแบบประเมินจากผู้ที่ได้ทดลองเล่นเกม ว่ามีความพึงพอใจกับตัวคู่ต่อสู้มากน้อยเพียงใด และมีความยากง่ายที่เหมาะสมหรือไม่
5. ทดสอบตัว Arm control system และเกม VR ที่พัฒนา วิจารณ์ว่าสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพหรือไม่
6. จัดทำแบบสำรวจเพื่อวัดผลความพึงพอใจหลังจากเล่นเกม VR ร่วมกับ Arm control system

## 7. แผนการดำเนินงาน (Gantt Chart)



รูป 20 แผนการดำเนินงาน

## 8. ประโยชน์ที่คาดว่าจะได้รับ (Expected Benefits)

1. เพิ่มอรรถรสในการเล่นเกมน เพิ่มความสนุก ทำให้เข้าถึงเกมมากยิ่งขึ้น
2. สามารถนำ Arm control system ไปประยุกต์ใช้กับเกม VR อื่นในอนาคตได้
3. เพื่อประยุกต์การทำ Machine Learning ลงในเกม
4. สามารถประยุกต์ใช้ในงานที่ต้องอาศัยการสัมผัสเสมือนจริงมาเกี่ยวข้องได้

## 9. ผลการศึกษาเทคโนโลยีที่ใช้พัฒนา

1. ส่วนของ Arm control system จากการศึกษาพบว่าแรงที่เกิดจาก servo motor มีค่อนข้างมากซึ่งถือว่าใช้ได้แต่เนื่องด้วยส่วนของแขนต้องการอิสระพอสมควรจึงมีปัญหาในส่วนนี้ซึ่งจะศึกษาว่าสามารถใช้ stepper motor ทดแทนได้หรือไม่
2. ส่วนของ MQTT ในการสื่อสารเพียง 1 node พบว่า latency ค่อนข้างน้อย แต่ยังไม่ได้ทดสอบรวมกันกับ Unity ซึ่งจะศึกษาเพิ่มเติมในเทอมหน้า
3. ส่วนของการศึกษา Unity พบว่า Unity สามารถพัฒนาเกมรูปแบบสามมิติ และสามารถใช้งานร่วมกับอุปกรณ์ VR โดยการติดตั้ง package ที่ชื่อว่า XR Interaction Toolkit ซึ่ง package นี้จะมีเครื่องมือสำหรับพัฒนาเกมที่ใช้เทคโนโลยี VR และ AR
4. ส่วนของการศึกษาตัว Unity ML-agent ที่เป็นตัว open-source จากทางผู้พัฒนาของ unity ได้ลองนำมาใช้ในการสร้าง AI ในลักษณะที่ให้ agent สามารถเดินเข้าไปหาเป้าหมายได้ โดยเป้าหมายจะมีการสุ่มเกิดไปเรื่อยๆ ก็สามารถทำได้ โดยตัว Tools นี้ไม่ได้ใช้งานยากจนเกินไป เนื่องด้วยมีอัลกอริทึมที่นิยมใช้ในการนำมาสร้าง AI ในเกมที่เป็นที่นิยม
5. Unity ML-agent สามารถที่จะสร้างกราฟออกมา เพื่อเปรียบเทียบการทำงานของแต่ละอัลกอริทึมได้ โดยใช้จากตัว Tensor Board ในการแสดงผลเปรียบเทียบประสิทธิภาพในการเทรนในแต่ละรอบ

## 10. เอกสารอ้างอิง (Reference)

- [1] MQTT from netpie introduction, [Online].Available :  
<https://netpie.gitbooks.io/doc/content/second-chapter.html>
- [2] Design and Development of Spinning Bike Game Based on VR Technology, [Online].Available :  
<https://ieeexplore.ieee.org/document/9177487>
- [3] Unity (game engine), [Online].Available : [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- [4] What is reinforcement learning? The complete guide, [Online]. Available:  
<https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/>
- [5] Unity ML-Agents Toolkit, [Online]. Available: <https://github.com/Unity-Technologies/ml-agents>
- [6] How Fast is MQTT? Statistical Model Checking and Testing of IoT Protocols, [Online].Available :  
<http://www.ist.tugraz.at/aichernig/publications/papers/qest18.pdf>
- [7] Design patterns applied for networked first person shooting game programming,  
 [Online].Available : <https://ieeexplore.ieee.org/document/6888715>
- [8] Applying Design Patterns in Game Programming, [Online].Available :  
<https://www.semanticscholar.org/paper/Applying-Design-Patterns-in-Game-Programming-Qu-Song/a5b818667e20bf97449427256825dffbd97d9556#related-papers>
- [9] Creating Pro-Level AI for Real-Time Fighting Game with Deep Reinforcement Learning,  
 [Online]. Available:[https://www.researchgate.net/publication/332301037\\_Creating\\_Pro-Level\\_AI\\_for\\_Real-Time\\_Fighting\\_Game\\_with\\_Deep\\_Reinforcement\\_Learning](https://www.researchgate.net/publication/332301037_Creating_Pro-Level_AI_for_Real-Time_Fighting_Game_with_Deep_Reinforcement_Learning)