เกมกระดานเชิงปฏิสัมพันธ์ที่ใช้เทคโนโลยีความเป็นจริงเสริม

INTERACTIVE BASED AUGMENTED REALITY BOARD GAME

รัฐพงศ์ อภิวัฒน์เจริญ

วชิรพล สถิรปัญญา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เกมกระดานเชิงปฏิสัมพันธ์ที่ใช้เทคโนโลยีความเป็นจริงเสริม

INTERACTIVE BASED AUGMENTED REALITY BOARD GAME

รัฐพงศ์ อภิวัฒน์เจริญ

วชิรพล สถิรปัญญา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เกมกระดานเชิงปฏิสัมพันธ์ที่ใช้เทคโนโลยีความเป็นจริงเสริม

INTERACTIVE BASED AUGMENTED REALITY BOARD GAME

ผู้จัดทำ

1. นายรัฐพงศ์ อภิวัฒน์เจริญ รหัสนักศึกษา 57011060
2. นายวชิรพล สถิรปัญญา รหัสนักศึกษา 57011085

_____ อาจารย์ที่ปรึกษา

(ผศ. ดร.สมศักดิ์ วลัยรัชต์)

_____ อาจารย์ที่ปรึกษาร่วม

(ผศ. ดร.อรัญญา วลัยรัชต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกมกระดานเชิงปฏิสัมพันธ์ที่ใช้เทคโนโลยีความเป็นจริงเสริม

นายรัฐพงศ์	อภิวัฒน์เจริญ	57011060
นายวชิรพล	สถิรปัญญา	57011085
ผศ. ดร.สมศักดิ์	วลัยรัชต์	อาจารย์ที่ปรึกษา
ผศ. ดร.อรัญญา	วลัยรัชต์	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2560

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์เพื่อประยุกต์ใช้เทคโนโลยี Augmented Reality (AR) ในรูปแบบของบอร์ดเกมจำลอง ด้วยเซนเซอร์ของ Leap Motion ซึ่งสามารถตรวจจับตำแหน่งในพิกัด 3 มิติของมือและนิ้วทั้งหมดได้ วัตถุเสมือนในเกม AR จะถูกควบคุมโดยท่าทางของมือและนิ้วของผู้เล่น และด้วย Google Cardboard headset สภาพแวดล้อม AR ของเกมจะถูกแสดงบนหน้าจอของสมาร์ทโฟนที่ใช้ระบบปฏิบัติการ Android ภาพรวมของระบบจะประกอบไปด้วย 3 ส่วนหลัก ส่วนแรกคือแอปพลิเคชันบนสมาร์ทโฟนสำหรับแสดงผลวัตถุเสมือน, user interface และสภาพแวดล้อม AR ของเกม ส่วนถัดไปคือ streamer ที่จะส่งตำแหน่งของมือในพิกัด 3 มิติจากเซนเซอร์ของ Leap Motion ไปยัง server และส่วนสุดท้ายคือ game server ที่จะรับผิดชอบในส่วนของการทำงาน และการควบคุมเกี่ยวกับการปฏิสัมพันธ์ทั้งหมดระหว่างผู้เล่นกับวัตถุเสมือน สำหรับการทดสอบระบบนั้น เราจะพัฒนาบอร์ดเกม AR ที่ชื่อว่า “The Lost Travelers” ที่ใช้เทคนิคและวิธีการดังที่กล่าวไว้ข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERACTIVE BASED AUGMENTED REALITY BOARD GAME

Mr. Rattaphong	Aphiwatcharoen	57011060
Mr. Wachirapol	Satirapunya	57011085
Asst.Prof.Dr. Somsak	Walairacht	Advisor
Asst.Prof.Dr. Aranya	Walairacht	Co-Advisor

Academic Year 2560

ABSTRACT

This project purposes a kind of Augmented Reality, AR, application in a form of virtual board game. With Leap Motion sensors, 3D positions of all fingers and both hands can be detected. Virtual objects in the AR game are controlled by hand and finger motions of the players. Using Google Cardboard headset, 3D AR environment of the game is displayed on the screen of android-based smartphone. The implemented system consists of 3 main parts. Firstly, smartphone application is for displaying the simulated objects, the user interface, and the AR game environment. Next part is the streamer that sends 3D coordinates of hand positions from Leap Motion sensor to the server. And the last part is the game server that responds for all processing and controlling all interactions performed by the players with the simulated objects. As the experiment, we develop an AR board game named "The Lost Travelers" using techniques and methods mentioned above.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับคำแนะนำ คำปรึกษาจากอาจารย์ที่ปรึกษา สมศักดิ์ วลัยรัชต์ และอาจารย์ที่ปรึกษาร่วมอรัญญา วลัยรัชต์ทั้งในเวลาและนอกเวลา รวมไปถึงการเสนอทางออกของปัญหา ทางคณะผู้จัดทำขอแสดงความขอบคุณเป็นอย่างสูง

ขอขอบคุณ Matthew Hallberg และ Jaihe Chen ที่เป็นแนวทางและตัวอย่างในการพัฒนาการใช้งาน Leap Motion กับ Vuforia SDK

ขอขอบคุณห้องวิจัย Information Security Advisory Group ที่เอื้อเฟื้ออุปกรณ์ และสถานที่ตลอด 24 ชม

รัฐพงศ์ อภิวัฒน์เจริญ

วชิรพล สถิรปัญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญภาพ VII

บทที่ 1 บทนำ 1

1.1 บทนำ 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ประโยชน์ที่คาดว่าจะได้รับ 2

1.4 ขอบเขตของโครงงาน 2

1.5 เนื้อหาของรายงาน 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 3

2.1 เทคโนโลยีความเป็นจริงเสริม 3

2.2 Unity 3D Engine 5

2.3 Leap Motion Modules 5

บทที่ 3 การออกแบบและพัฒนา 7

3.1 ที่มาของการออกแบบ 7

3.2 การออกแบบฮาร์ดแวร์ 9

3.3 การออกแบบซอฟต์แวร์ 11

3.4 การออกแบบเกมกระดาน The Lost Travelers 18

3.5 Use case diagram 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 การทดลองและผลการทดลอง 27
4.1 โปรแกรมสำหรับส่งข้อมูล Leap Motion 27
4.2 แอปพลิเคชันบนสมาร์ทโฟน 30
4.3 Server 39

บทที่ 5 บทสรุปและข้อเสนอแนะ 44
5.1 ผลการดำเนินงาน 44
5.2 ปัญหาและอุปสรรค 44
5.3 แนวทางในอนาคต 45

บรรณานุกรม 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

2.1 Vuforia 6.2.10 Prefabs 4

3.1 รายละเอียดของการ์ดไอเทม 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูป	หน้า
2.1 ตัวอย่างเทคโนโลยีความเป็นจริงเสริม	3
2.2 ตัวอย่าง VuMark	4
2.3 หน้าตาโปรแกรม Unity 3D	5
2.4 มุมมองจาก Sensor Leap Motion	5
2.5 ตัวอย่างการใช้งาน Leap Motion Modules	6
3.1 ตำแหน่งการวางอุปกรณ์	7
3.2 HMD ที่มีการติดตั้ง Leap Motion	8
3.3 การแสดงผลมือก่อนการ Calibrate	8
3.4 การส่งข้อมูลจาก Leap Motion ไปยังสมาร์ทโฟน	9
3.5 การทำงานระหว่างสมาร์ทโฟนกับ Server	9
3.6 Port Diagram	10
3.7 System Block Diagram	10
3.8 รูปแบบข้อความสำหรับตำแหน่งมือ	11
3.9 ตัวอย่างข้อความตำแหน่งมือ	11
3.10 การทำงานบน Server	12
3.11 UDP Handshake	13
3.12 รูปแบบข้อความสำหรับการแสดงผล	13
3.13 รูปแบบข้อความสำหรับ Navigator	14
3.14 รูปแบบข้อความสำหรับ Navigator	14
3.15 รูปแบบข้อความสำหรับตำแหน่งหัว	15
3.16 การอ้างอิงตำแหน่งของหัวและมือของแต่ละโปรแกรม	15
3.17 Interaction Flowchart	17
3.18 แผนที่ ระดับความสูงของช่องในภูมิประเทศต่าง ๆ และการวางตัวละครบนช่อง	18
3.19 วัตถุต่าง ๆ ที่วางบนแผนที่	19
3.20 การจัดวางสภาพแวดล้อมของเกม	19
3.21 ตัวอย่างการเดินจากช่องหนึ่งไปยังอีกช่องหนึ่ง	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูป **หน้า**

3.22 วิธีการเดินของสัตว์ป่า 22
3.23 Use case Diagram 24
3.24 Activity Diagram 25
3.25 Class Diagram 26
4.1 การตั้งค่า Hand Controller Script 27
4.2 hierarchy ของโมเดลมือ 29
4.3 หน้าต่าง UI ของ Streamer 30
4.4 Vuforia License Manager 37
4.5 Vuforia Target Manager 38
4.6 Vuforia AR 38
4.4 หน้าต่าง UI ของ Server 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 บทนำ

เทคโนโลยีความเป็นจริงเสริม (Augmented Reality) เป็นเทคโนโลยีที่สามารถแสดงวัตถุเสมือนในสภาพแวดล้อมจริงในรูปแบบต่าง ๆ เช่น เสียง ภาพ วิดีโอ เป็นต้น ในปัจจุบันแล้วเทคโนโลยีความเป็นจริงเสริมเริ่มมีบทบาทและถูกนำมาประยุกต์ใช้ในหลาย ๆ ด้าน ไม่ว่าจะเป็น การโฆษณา การเรียนการสอน โดยเฉพาะอย่างยิ่ง การนำมาประยุกต์ใช้กับเกม ยกตัวอย่างเช่น Ingress , Pokémon Go , Night Terrors ซึ่งจะให้ผู้เล่นดำเนินเกมโดยปฏิสัมพันธ์กับวัตถุ หรือสถานที่ในโลกความเป็นจริง และจะมีการตอบสนองเป็นวัตถุในโลกเสมือน ยกตัวอย่างเช่น ผู้เล่นที่เล่นเกม Pokémon Go ต้องเดินทางไปยังสถานที่ต่าง ๆ เพื่อจับ Pokémon เป็นต้น กลุ่มผู้จัดทำจึงต้องการนำเทคโนโลยีความเป็นจริงเสริมมาสร้างเป็นเกมในรูปแบบของเกมกระดาน ซึ่งกำลังเป็นที่นิยมในปัจจุบันเพราะเหมาะสมกับทุกเพศทุกวัย และยังเปิดโอกาสให้ผู้เล่นได้มีปฏิสัมพันธ์โดยตรงกับผู้เล่นคนอื่น ๆ อีกด้วย

แม้ว่าเทคโนโลยีความจริงเสริมจะทำให้เกิดการปฏิสัมพันธ์ระหว่างผู้เล่นกับโลกความเป็นจริงแล้ว แต่ก็ยังมีข้อจำกัดในด้านการควบคุม ดังนั้นแล้วกลุ่มผู้จัดทำจึงได้นำ Leap Motion ซึ่งเป็นอุปกรณ์สำหรับตรวจจับตำแหน่งในพิกัด 3 มิติ และท่าทางหรือการเคลื่อนไหวของมือ มาใช้เป็นอุปกรณ์สำหรับให้ผู้ใช้สามารถออกคำสั่งต่าง ๆ ในเกมซึ่งจะช่วยเพิ่มความหลากหลายในการควบคุม และยังเพิ่มความรู้สึกสมจริงกับวัตถุเสมือนอีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาเกี่ยวกับเทคโนโลยีความเป็นจริงเสริม และการประยุกต์ใช้งานในชีวิตจริง
2) เพื่อศึกษาการทำงานของ Leap Motion และการประยุกต์ใช้งานในชีวิตจริง
3) เพื่อศึกษาเกี่ยวกับการเขียนโปรแกรมเพื่อสร้างเกมคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้รับความรู้เกี่ยวกับเทคโนโลยีความเป็นจริงเสริม และการประยุกต์ใช้งานในชีวิตจริง
2) ได้รับความรู้เกี่ยวกับการทำงานของ Leap Motion และการประยุกต์ใช้งานในชีวิตจริง
3) ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมเพื่อสร้างเกมคอมพิวเตอร์
4) สามารถนำความรู้ที่ได้รับไปต่อยอดพัฒนาในด้านต่าง ๆ เช่น การศึกษา การบำบัด การเสริมสร้างพัฒนาการ ความสะดวกสบาย เป็นต้น

1.4 ขอบเขตของโครงงาน

1) สร้างสภาวะความเป็นจริงเสริมให้ผู้เล่นทั้งสองสามารถมองเห็นร่วมกันได้
2) ตรวจจับท่าทาง ตำแหน่ง และลักษณะของมือทั้งสองของผู้เล่น
3) นำท่าทาง ตำแหน่ง และลักษณะของมือของผู้เล่น มาแปลเป็นคำสั่ง
4) ใช้คำสั่งจากผู้เล่นมาดำเนินเกมและแสดงผลในรูปแบบความเป็นจริงเสริมให้แก่ผู้เล่นทั้งสอง

1.5 เนื้อหาของรายงาน

บทที่ 1 บทนำ กล่างถึงวัตถุประสงค์ของโครงงาน ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของโครงงาน บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงเทคโนโลยีความเป็นจริงเสริม Vuforia Software Development, Leap Motion Unity Module, Unity 3D Engine บทที่ 3 การออกแบบและพัฒนา กล่าวถึงการออกแบบด้านฮาร์ดแวร์ การออกแบบด้านซอฟต์แวร์ และการออกแบบเกมกระดาน บทที่ 4 การดำเนินงานและผลการดำเนินงาน กล่าวถึงการทดลองใช้งานโปรแกรม Unity 3D การทดลองส่งข้อมูลระหว่าง PC กับสมาร์ทโฟน การทดลองใช้งาน Vuforia SDK บทที่ 5 บทสรุป กล่าวถึงผลการดำเนินการ ปัญหาอุปสรรค และแนวทางการพัฒนาระบบในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

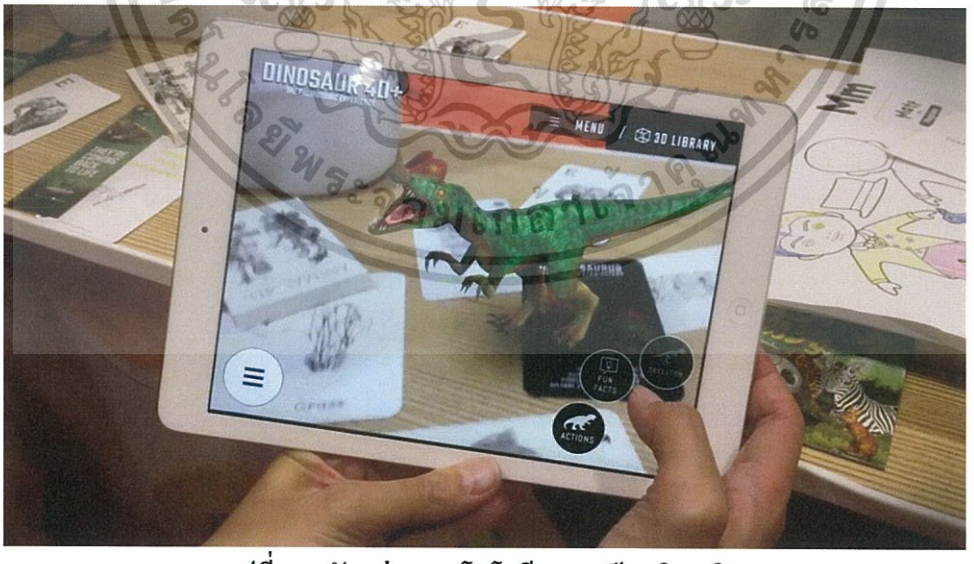
บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 เทคโนโลยีความเป็นจริงเสริม

เทคโนโลยีความเป็นจริงเสริม (Augmented Reality) เป็นเทคโนโลยีที่แสดงข้อมูลจำลองหรือข้อมูลที่สร้างขึ้นในโลกจริง ผ่านหน้าจอแสดงผลของอุปกรณ์แบบไร้รอยต่อ โดยอาจจะมีการรับข้อมูลมาจากโลกจริงในบางครั้ง ซึ่งถูกแบ่งออกเป็น 2 แบบได้แก่ การแสดงผลแบบไม่ใช้สัญลักษณ์ กับการแสดงผลแบบมีภาพที่เป็นสัญลักษณ์หรือตำแหน่งที่แสดงผลของมูลเสมือนจริง นิยมเรียกว่า "Marker" หรืออาจจะเรียกว่า AR Code ก็ได้ ในขั้นแรกกล้องจะทำการรับภาพเข้ามา แล้วประมวลผลด้วยซอฟต์แวร์ เมื่อเจอสัญลักษณ์หรือ Marker ซอฟต์แวร์จะแสดงข้อมูลจำลองในตำแหน่งเดียวกับสัญลักษณ์

2.1.1 องค์ประกอบของเทคโนโลยีความเป็นจริงเสริม

1) สัญลักษณ์หรือ Marker ใช้ในการระบุตำแหน่งที่จะแสดงผลข้อมูลจำลอง
2) กล้องหรือ Sensor ที่ใช้ในการรับข้อมูลภาพแล้วส่งต่อให้ซอฟต์แวร์เพื่อประมวลผล
3) AR Engine หรือซอฟต์แวร์ที่ทำหน้าที่ประมวลผลภาพที่ได้รับมาแล้วเพื่อแสดง
4) ส่วนแสดงผล แสดงภาพหลังประมวลผลด้วย AR Engine ที่อาจจะมีข้อมูลจำลองอยู่



รูปที่ 2.1 ตัวอย่างเทคโนโลยีความเป็นจริงเสริม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Vuforia

SDK (Software Development Kit) สำหรับการพัฒนาในส่วนการประมวลผลตำแหน่ง Marker สามารถตรวจจับประเภทของ Marker ได้หลายรูปแบบ ไม่ว่าจะเป็น รูปภาพ หรือวัตถุรูปทรงต่าง ๆ นอกจากนี้ยังรองรับการทำ Virtual Buttons, การสร้างแผนที่ 3 มิติ ด้วย Smart Terrain ออกแบบมาให้ใช้งานบนระบบปฏิบัติการ Android, iOS และ UWP แอปพลิเคชัน

ตาราง 2.1 Vuforia 6.2.10 Prefabs

Prefabs	**การใช้งาน**
ARCamera	กล้องและการตั้งค่า Vuforia Behavior ทั้งหมด
CloudRecognition	สำหรับการจัดการ target ที่อยู่บน Cloud ที่ต้องมีการอัพเดทบ่อยครั้ง
CylinderTarget	สำหรับ defines target ที่มีรูปร่างเป็นทรงกระบอก
ImageTarget	สำหรับ defines target ที่เป็นรูปภาพ
MultiTarget	สำหรับ defines target ที่เป็นรูปภาพหลายๆรูป
ObjectTarget	สำหรับ defines target ที่เป็นสิ่งของขนาดไม่ใหญ่มาก เช่น รถของเล่น
TextRecognition	สำหรับ defines target ที่เป็นตัวอักษรภาษาอังกฤษ
UserDefinedTargetBuilder	สำหรับ defines target ที่ผู้ใช้ defines ขณะ runtime
VirtualButton	ตัวช่วยในการตั้งค่าให้ object ที่แสดงบน target มีความสามารถ interact กับ object อื่น
VuMark	สำหรับ defines target ที่เป็น custom barcode
Word	ใช้คู่กับ TextRecognition สำหรับการ defines target ที่เป็นประโยค

รูปที่ 2.2 ตัวอย่าง VuMark

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

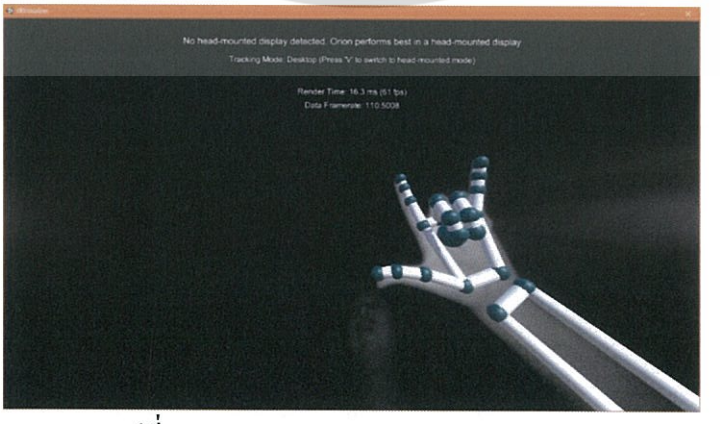
2.2 Unity 3D Engine

โปรแกรมที่ใช้ในการพัฒนาซอฟต์แวร์เกมเอนจิน หรือเกมแอนิเมชัน สามาถสร้างเกมได้ทั้ง 2 มิติ และ 3 มิติ สามารถทำงานร่วมกับ Vuforia ได้ โดยโปรแกรมมีจุดเด่นในการทำงานเป็นการจัดตัว Object แบบ Hierarchy มี Asset Store ให้โหลดส่วนเสริม โปรแกรมนี้ใช้ภาษา C#, JAVA เป็นหลัก และสามารถ Export ไฟล์ออกมาเป็น .apk ได้อีกด้วย

รูปที่ 2.3 หน้าตาโปรแกรม Unity 3D

2.3 Leap Motion Modules

เป็นโมดูลสำหรับนำข้อมูลที่ได้จาก Leap Motion มาใช้งานใน Unity ซึ่งจะช่วยให้แอพพลิเคชั่นที่สร้างสามารถใช้งาน Leap Motion ได้ โดยในโครงงานนี้จะใช้โมดูลพื้นฐานของ Leap Motion ที่จะช่วยในการนำข้อมูลที่ได้จาก Leap Motion มาใช้งานในแอพพลิเคชั่นโดยจะมีส่วนประกอบต่าง ๆ ดังนี้



รูปที่ 2.4 มุมมองจาก Sensor Leap Motion

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 LeapServiceProvider

ทำหน้าที่ในการติดต่อกับ Leap Service แล้วนำ Frame Object ที่เก็บข้อมูลของมือมาให้กับแอพพลิเคชั่น

2.3.2 LeapHandController

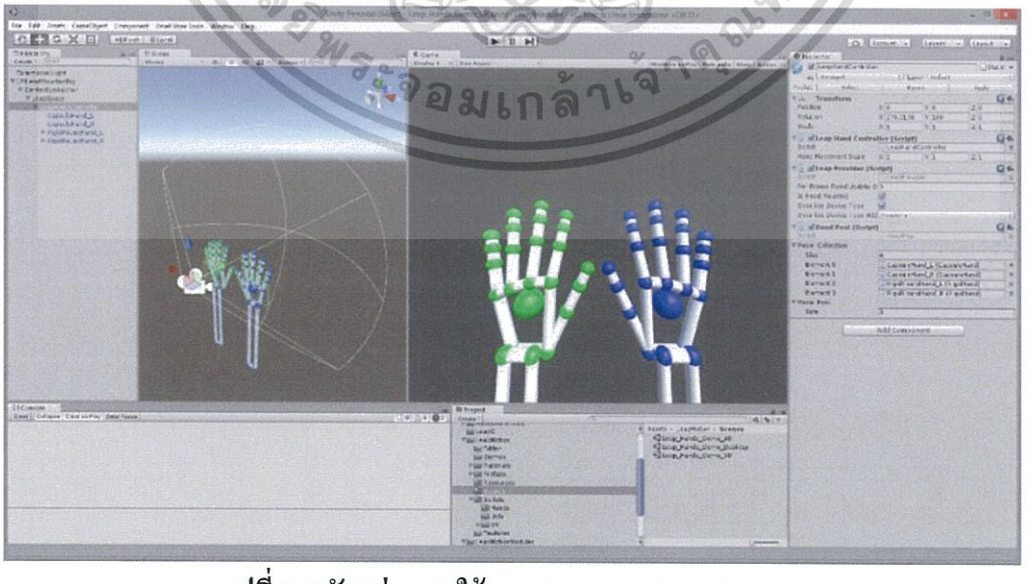
ทำหน้าที่ในการนำ Fame Object ที่ได้รับจาก LeapServiceProvider มาสร้าง หรืออัพเดทโมเดลของมือในแอพพลิเคชั่น ซึ่งจะทำงานแบบเฟรมต่อเฟรม โดยจะต้องมี HandPool อยู่ใน Object เดียวกันเพื่อที่จะใช้งาน

2.3.3 HandPool

เป็นตัวดำเนินการกับกลุ่มของข้อมูลของมือ ซึ่งจะถูกใช้โดย LeapHandController ในการนำข้อมูลกลับมาใช้ใหม่ โดยจะต้องเพิ่มโมเดลของมือลงไปในพารามิเตอร์เพื่อนำโมเดลนั้น ๆ มาใช้

2.3.4 LMHeadMountedRig

เป็นตัวช่วยสำหรับการสร้างแอพพลิเคชั่นที่ใช้ Leap Motion ในรูปแบบของ Virtual Reality โดยประกอบไปด้วยส่วนหลัก ๆ สามส่วน ส่วนแรก คือ CenterEyeAnchor ทำหน้าที่เป็น Main Camera ซึ่งตำแหน่งและการหมุนต่าง ๆ จะถูกควบคุมโดยตรงจาก Unity's VR integration ทำให้ไม่สามารถเลื่อนตำแหน่งได้โดยตรง และมี LeapSpace เป็นส่วนประกอบ ซึ่งจะทำหน้าที่ปรับเปลี่ยนตำแหน่งและการหมุนของมือให้เข้ากับมุมมองของ VR โดยจะมี LeapHandController เป็นส่วนประกอบของ LeapSpace ส่วนที่สอง HandModel เป็นโมเดลของมือที่จะใช้ในแอพพลิเคชั่น และส่วนสุดท้าย Object อื่น ๆ ที่ผูกติดกับตัวผู้เล่น



รูปที่ 2.5 ตัวอย่างการใช้งาน Leap Motion Modules

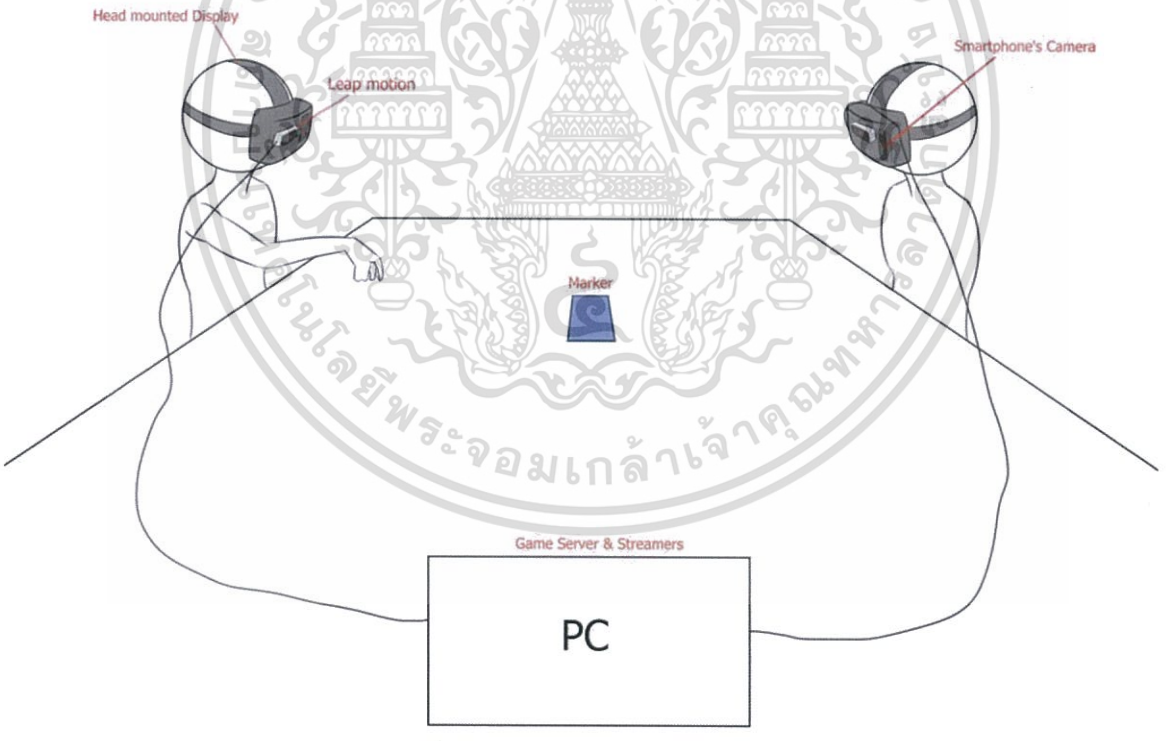
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและพัฒนา

3.1 ที่มาของการออกแบบ

การออกแบบนั้นเราจะให้เป็นเกมสำหรับผู้เล่นสองคน เพราะฉะนั้นแล้ว เราจึงต้องใช้พื้นที่ประมาณ 40x50 ตารางเซนติเมตร ในการวางอุปกรณ์ทั้งหมด ให้ผู้เล่นทั้งสองคนนั่งในทิศตรงข้ามกันมีแผ่น Mark ที่เป็น Image Target อยู่ตรงกลางขนาด 28x28 ตารางเซนติเมตร มี PC วางอยู่ด้านข้างที่รันโปรแกรม Streamer, Server และเชื่อมต่อกันบน Local Wi-Fi ผู้เล่นทั้งสองคนจะสวม HMD ที่มีการติดตั้งของ Leap Motion ไว้ด้านหน้า มีการวาง Smartphone รันแอพพลิเคชัน Client ที่มีการแสดงผลเป็น split screen 2 จอไว้ด้านในของ HMD ดังรูปที่ 3.1

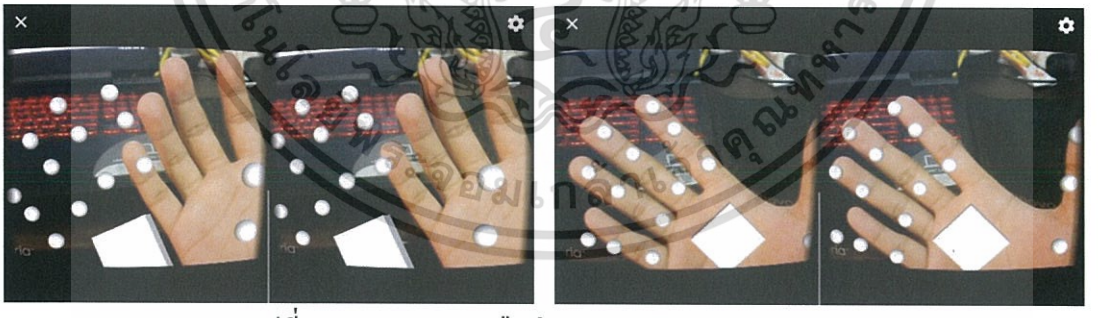


รูปที่ 3.1 ตำแหน่งการวางอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 HMD ที่มีการติดตั้ง Leap Motion

แต่เนื่องด้วยตำแหน่งกล้อง Leap Motion ที่ตรวจจับมือ กับ กล้องของ Smartphone ที่ใช้เป็น Video Background นั้นไม่ได้อยู่ในตำแหน่งเดียวกัน ทำให้ภาพที่ออกมามีความคลาดเคลื่อนประมาณ 2-3 เซนติเมตร ขึ้นอยู่กับการติดตั้ง โดยการแสดงผลบน Smartphone จะคลาดเคลื่อนตามไปด้วยดังรูปที่ 3.3



รูปที่ 3.3 การแสดงผลมือก่อนและหลังการ Calibrate

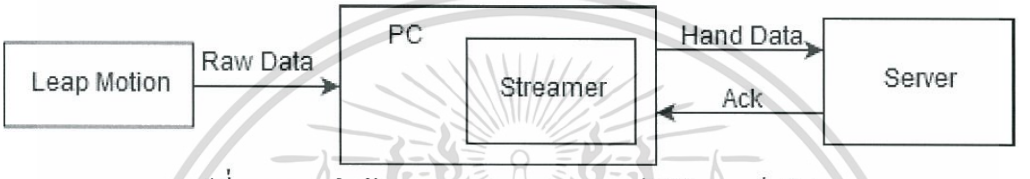
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบฮาร์ดแวร์

ในส่วนของการออกแบบฮาร์ดแวร์นั้นได้ถูกแบ่งออกเป็น 3 ส่วนดังนี้

3.2.1 การสตีมมิ่งข้อมูล Leap Motion

เนื่องด้วย Leap Motion ไม่รองรับการส่งข้อมูลระหว่างสมาร์ทโฟน จึงจำเป็นต้องใช้ PC เป็นตัวช่วยในการส่งข้อมูลตำแหน่งพิกัดมือจาก Leap Motion ไปยังสมาร์ทโฟน เพื่อให้สมาร์ทโฟนประมวลแล้วใช้ในการแสดงผล โดยข้อมูลตำแหน่งพิกัดมือของ Leap Motion จะถูกส่งผ่าน Wi-Fi แบบ UDP โดยมีการกำหนด IP และ Port ไว้ ทั้งนี้ทั้งนั้นแล้วสมาร์ทโฟนและ PC ต้องอยู่ในเครือข่ายเดียวกัน



รูปที่ 3.4 การส่งข้อมูลจาก Leap Motion ไปยังสมาร์ทโฟน

3.2.2 การทำงานบนสมาร์ทโฟน

บนสมาร์ทโฟนจะมีการทำงานของกล้องเพื่อรับข้อมูลมาประมวลผลของความเป็นจริงเสริม เพื่อแสดงผลและส่งตำแหน่งของกล้องที่ใช้แทนตำแหน่งของหัว ไปให้ Server หลังจากที่ได้รับข้อมูลจาก Server การแสดงผลบนสมาร์ทโฟนนั้นจะมีการแสดงผลมือที่ Leap Motion จับได้ ความเป็นจริงเสริมที่ได้รับจากกล้อง ทั้งหมดนี้จะถูกแสดงให้อยู่ในรูปแบบของ Virtual Reality Screen

รูปที่ 3.5 การทำงานระหว่างสมาร์ทโฟนกับ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

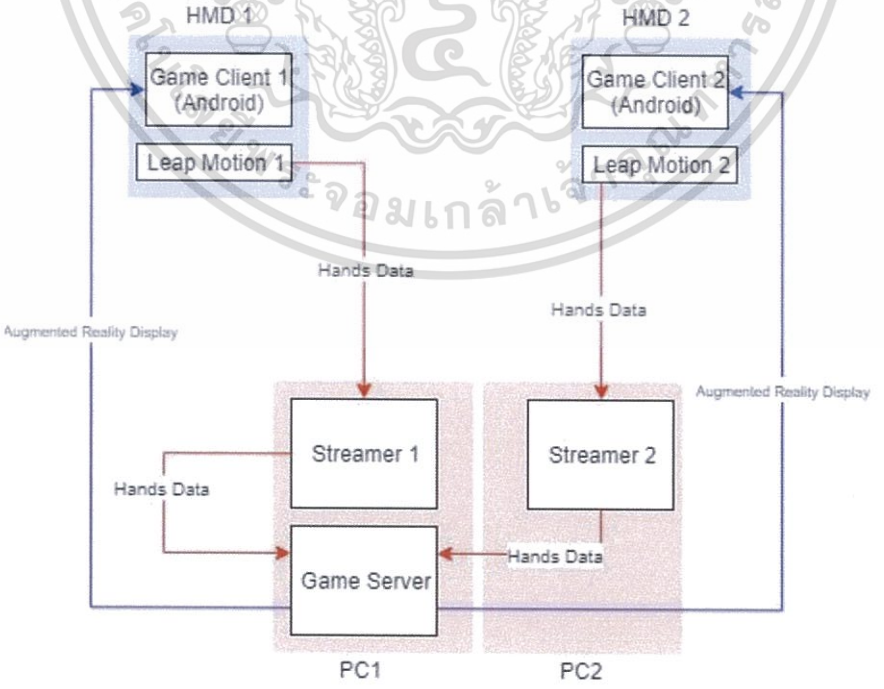
3.2.3 การทำงานบน Server

บน Server ก็จะมีหน้าที่ดำเนินเกมหรือเป็นตัวกลางที่คอยให้ผู้เล่นแรกติดต่อกับผู้เล่นคนที่สองได้ เพื่อให้ข้อมูลมีความสัมพันธ์กัน โดยการติดต่อสื่อสารระหว่างโปรแกรมนั้น เราจะกำหนดให้ Port 1999 ใช้สำหรับข้อมูลที่ส่งมายัง Server และ Port 1998 สำหรับข้อมูลที่ส่งออกจาก Server ไปให้ Client และ Streamer

รูปที่ 3.6 Port Diagram

3.2.4 การทำงานของระบบ

สำหรับผู้ใช้งานหนึ่งคนนั้นจะประกอบไปด้วย PC, Smartphone และ Leap Motion ซึ่งบน PC จะมีการต่อ Leap Motion ไว้ และเปิดการใช้งานโปรแกรม Streamer หากเป็น Host ต้องเปิดโปรแกรม Server ด้วย มีการทำงานของแอพ Client บน Smartphone และถูกติดตั้งไว้บน HMD ควบคู่กับ Leap Motion เมื่อมีการสัมผัสหรือการ Interaction ใด ๆ เกิดขึ้น ข้อมูลจะถูก Broadcast ไปยัง Client ทั้งสอง เพื่อให้มองเห็นสภาพแวดล้อร่วมกัน



รูปที่ 3.7 System Block Diagram

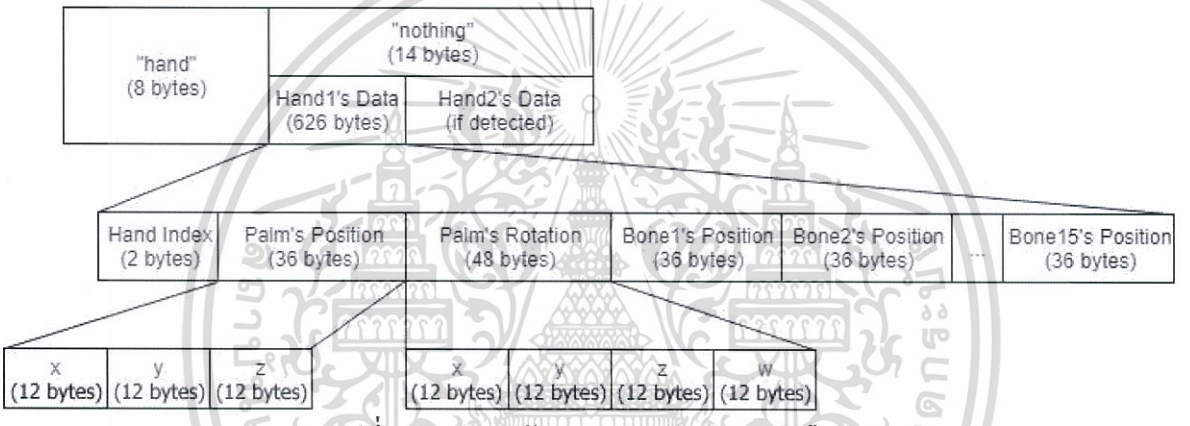
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบซอฟต์แวร์

ในส่วนของการออกแบบฮาร์ดแวร์นั้นได้ถูกแบ่งออกเป็น 3 ส่วนดังนี้

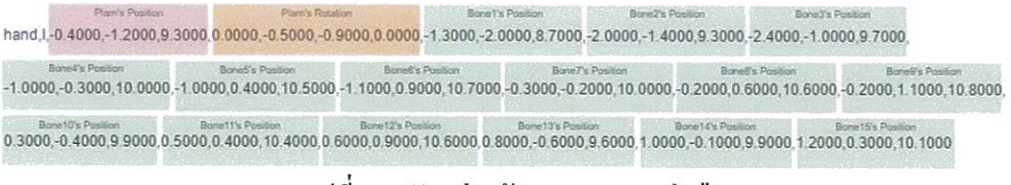
3.3.1 ซอฟต์แวร์การส่งข้อมูล Leap Motion

ซอฟต์แวร์จะคอยทำหน้าที่รับข้อมูลที่ Leap Motion อ่านได้แล้วส่งตำแหน่งพิกัดของมือที่อ้างอิงจากตำแหน่งของ Leap Motion ไปให้ซอฟต์แวร์บนสมาร์ทโฟนเพื่อทำการประมวลผลต่อไป การส่งข้อมูลจะถูกส่งแบบ UDP ผ่าน Wi-Fi ในเครือข่ายเดียวกัน นอกจากนี้ซอฟต์แวร์ยังมีหน้าที่คอย Calibrate ให้ตำแหน่งมือตรงกับตำแหน่งที่ควรจะอยู่โดยการคำนวณค่า Offset เข้าไปก่อนส่งไปยัง Server โดยข้อมูลจะถูกส่งไปในรูปแบบ Array of Byte มี Packet ดังภาพที่ 3.8



รูปที่ 3.8 รูปแบบข้อความสำหรับตำแหน่งมือ

โดยหลังจาก Decode ข้อความแล้วนั้น “hand” คือประเภทของข้อความที่บอกว่าข้อความนี้เป็นข้อมูลมือ “l” หรือ “r” คือ ประเภทของมือว่าเป็นมือซ้ายหรือมือขวา Palm Position คือ ตำแหน่งของจุดศูนย์กลางมือ เป็นข้อมูลประเภท Vector3 Palm Rotation คือ ลักษณะการหมุนของจุดศูนย์กลางมือเป็นข้อมูลประเภท Quaternion และ Finger bone Position คือ ตำแหน่งของกระดูกข้อต่อต่าง ๆบนมือเป็นข้อมูลประเภท Vector3 ในกรณีที่ Leap Motion ไม่พบมือจะมีการข้อมูลไปยัง Server เช่นกัน แต่ว่า Message นั้นจะแตกต่างออกไป โดยจะตามด้วย “nothing”

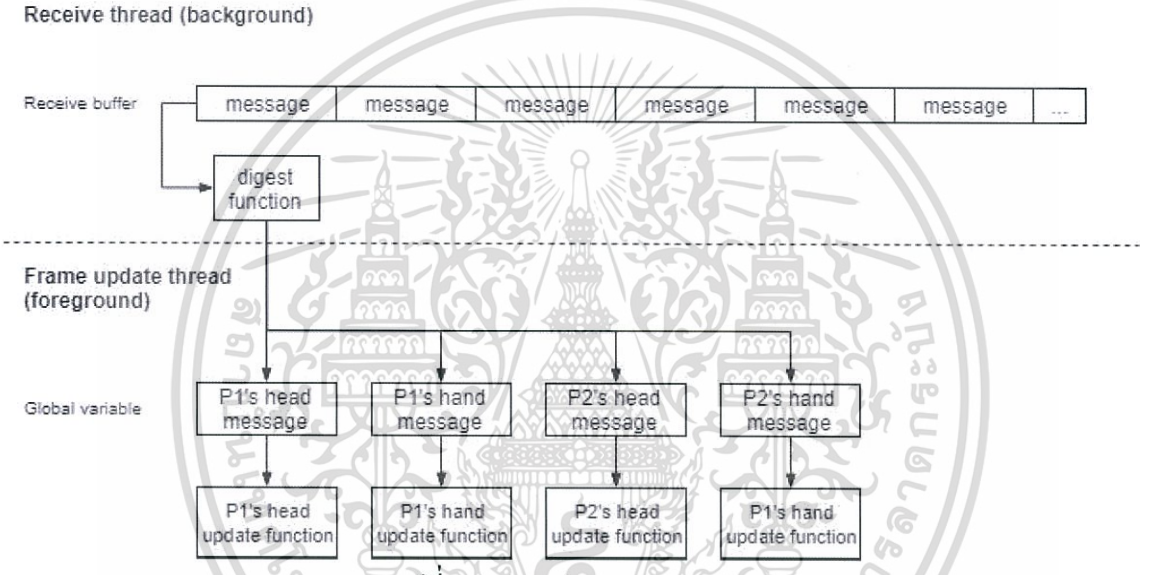


รูปที่ 3.9 ตัวอย่างข้อความตำแหน่งมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ซอฟต์แวร์ที่ทำหน้าที่เป็น Server

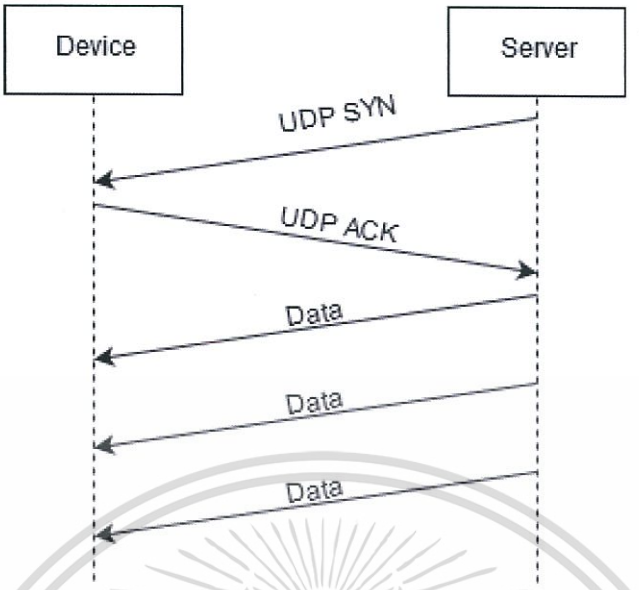
ซอฟต์แวร์นี้จะทำหน้าที่อัพเดทข้อมูลระหว่างผู้เล่นให้มีความสัมพันธ์กัน เช่น หากผู้เล่นได้ใช้คำสั่งเดิน Client จะส่งข้อมูลไปอัพเดทที่ Server เพื่อให้ผู้เล่นอีกคนหนึ่งอัพเดทข้อมูลการเดินของผู้เล่นคนแรก นอกจากนี้การดำเนินเกมต่าง ๆ ก็ถูกจัดการภายในซอฟต์แวร์นี้ มีหน้าที่หลัก ๆ คือ รับ Message มา ประมวลผล และส่งให้ Client หากมีการอัพเดทการแสดงผล โดยมี Background Thread เป็นส่วนในการรับ UDP Message ที่รับจาก Port 1999 ไปเก็บไว้ใน Queue จากนั้น Foreground Thread จะดึงข้อมูลใน Queue มา Decode และ Execute



รูปที่ 3.10 การทำงานบน Server

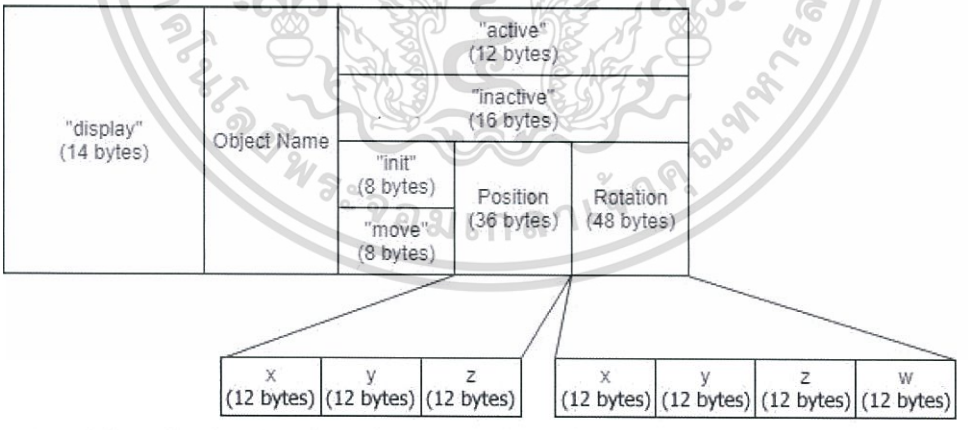
สำหรับเครื่องที่เป็น Host แล้วนั้น ต้องกรอก IP ลงบน Server เพื่อให้ Client และ Streamer สามารถรู้จักแล้วติดต่อกันได้ โดย Server จะส่งข้อความไป Handshake ดังรูปที่ 3.9 และเมื่อได้ข้อความตอบกลับ จะเข้าสู่สถานะพร้อมใช้งาน หลังจากนั้น Server จะส่งข้อความการแสดงผลไปให้กับ Client และรับข้อมูลตำแหน่งหัวกับมือในเวลาเดียวกัน ข้อมูลบนที่เก็บบน Server จะประกอบไปด้วย IP Address ของ Client และ Streamer ตำแหน่งมือ ตำแหน่งหัว ตำแหน่ง Object ต่าง ๆ ที่ใช้สำหรับการแสดงผล ข้อมูลผู้เล่น และ flag ต่าง ๆ หากตำแหน่งของ Object มีการเปลี่ยนแปลงหรือขยับก็จะ Broadcast ให้กับ Client ทั้งสองเพื่อที่จะได้เห็นสิ่งแวดล้อมร่วมกัน และในกรณีที่ device มีการปิดตัวหรือขาดการเชื่อมต่อ จะมีการส่งข้อความมาที่ Server ว่า “disconnect” หลังจากนั้น Server จะ reset state กลับสู่สถานะ pending แล้ว broadcast ข้อความต่อไปให้ device ที่เหลือ reset สู่สถานะ pending เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 UDP Handshake

ข้อความประเภท “display” เป็นข้อความที่ใช้จัดการการแสดงผลทั้งหมด โดย Object Name คือชื่อของ Object ที่อ้างอิง “init” คือ คำสั่ง generate object สำหรับ object ที่ไม่เคยมีอยู่จะถูกสร้างขึ้น “active” คือ แสดงผล “inactive” คือ ซ่อนการแสดงผล และ “move” สำหรับย้าย Object ที่อ้างอิงไปยังตำแหน่งนั้น ๆ



Ex: display,rock12,init,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000,0.0000
display,player1,move,1.0000,1.0000,1.0000,0.0000,0.0000,0.0000,0.0000
display,forest1,inactive

รูปที่ 3.12 รูปแบบข้อความสำหรับการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

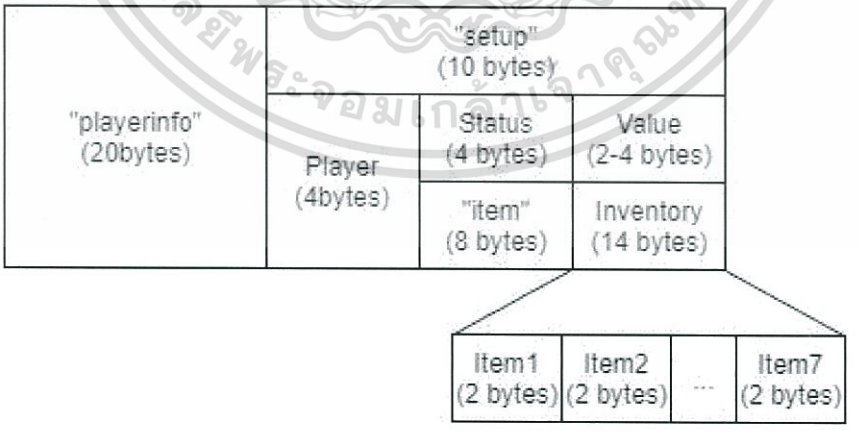
ในส่วนของการแสดงผล Navigator จะมีการส่งข้อความออกจาก Server โดยขึ้นต้นข้อความว่า “navigator” และตามด้วย “turninfo” เป็นการบอกข้อมูลรอบ Round เป็นรอบที่ และ Turn คือ ตาเดินของผู้เล่น หากตามด้วย “gameover” จะถูกส่งออกจาก Server ในกรณีที่เกมได้จบลง โดยจะตามด้วยข้อมูลผู้เล่นทั้งสองคนว่าแพ้หรือชนะ “w” แทนชนะ และ “l” แทนแพ้

"navigator" (18 bytes)	"keyitem" (14 bytes)	Key Item (6-8 bytes)	
	"turninfo" (16 bytes)	Round (2 bytes)	Turn Phase (4-14 bytes)
	"gameover" (16 bytes)	P1's Result (2 bytes)	P2's Result (2 bytes)

Ex: navigator,keyitem,log
navigator,turninfo,2,flood
navigator,gameover,w,l

รูปที่ 3.13 รูปแบบข้อความสำหรับ Navigator

นอกจากนั้นยังมีข้อความที่อัพเดทการแสดงผลค่าสถานะต่าง ๆ ของตัวละครผู้เล่นใน Player’s Area โดยจะเริ่มต้นข้อความด้วย “playerinfo” ตามด้วย Status คือ ค่าสถานะต่าง ๆ ของผู้เล่น เช่น HP, SP, TP เป็นต้น และ Value คือ ค่าสถานะนั้น ๆ เป็นตัวเลข ถ้าหากข้อความ “playerinfo” ตามด้วย “setup” ข้อความนั้นจะมีความหมายว่า ให้ Reset สถานะทุกค่ากลับสู่ค่าเริ่มต้น



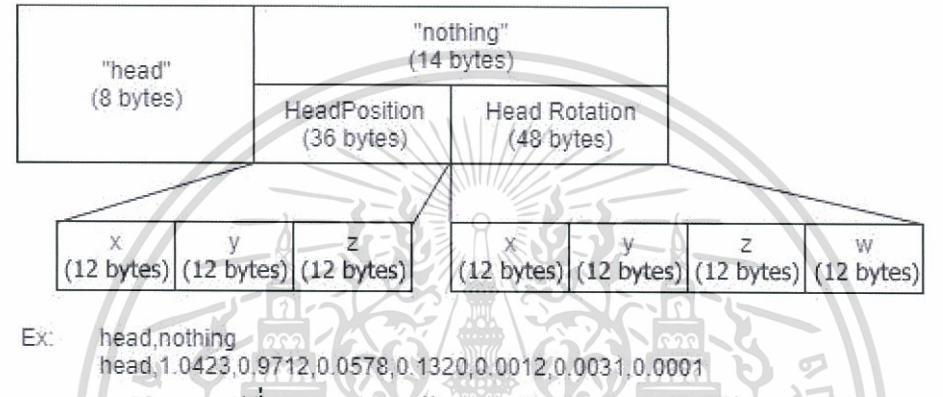
Ex: playerinfo,setup
playerinfo,p1,hp,5
playerinfo,p2,item,1,3,1,0,n,n,n

รูปที่ 3.14 รูปแบบข้อความสำหรับ Navigator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ซอฟต์แวร์แสดงผลบนสมาร์ทโฟน

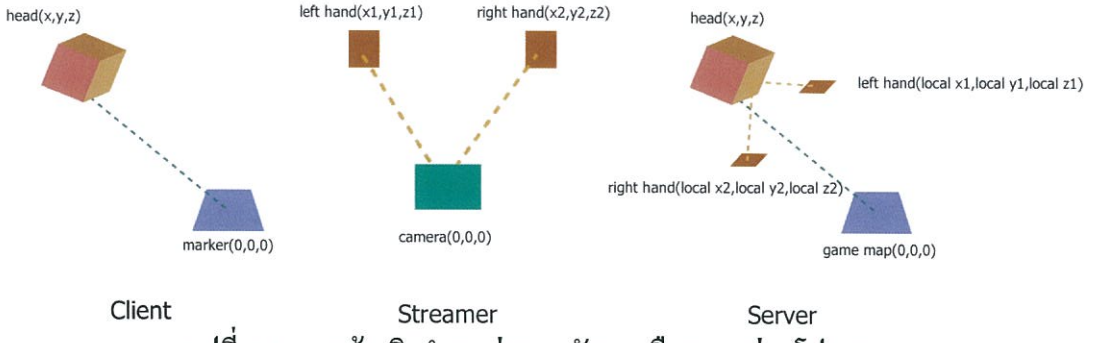
ซอฟต์แวร์บนสมาร์ทโฟนมีหน้าที่ส่งตำแหน่งพิกัดของกล้องเมื่อจับภาพของ Image target หรือ mark ได้ โดยอ้างอิงจากจุดนั้น และแสดงผลข้อมูลทุกอย่างให้ ผู้เล่นเห็นในรูปแบบ VR Screen โดยข้อมูลที่จะถูกส่งมาที่ซอฟต์แวร์นี้มี ตำแหน่งพิกัด Object ที่แสดงผล ข้อมูลอัพเดทตัวเกมจาก Server นอกจากนี้ซอฟต์แวร์ยังต้องทำหน้าที่ประมวลผล AR โดยใช้ Unity Asset ที่มีชื่อว่า Vuforia เป็นตัวช่วยในการประมวลผลตำแหน่งของ AR Object



รูปที่ 3.15 รูปแบบข้อความสำหรับตำแหน่งหัว

3.3.4 การหาตำแหน่งมือในโลกของความเป็นจริง

สำหรับการ Interact นั้น Server จำเป็นต้องรู้ตำแหน่งของมือและตำแหน่งของหัวเพื่อที่จะสามารถจำลองออกมาเป็นตำแหน่งในโลกจริงได้โดย จุดศูนย์กลางหรือจุดอ้างอิงคือ Image target ที่ x, y, z เท่ากับ 0, 0, 0 โดยที่ตำแหน่งหัวจะรับมาจาก Client ซึ่งใช้จุดอ้างอิงแบบเดียวกับ Server และตำแหน่งมือจะรับมาจาก Streamer ซึ่งใช้ Camera ของโปรแกรมเป็นจุดอ้างอิง จากนั้นจึงจัดเรียง Heirachy ของหัวและมือให้มือนั้นเป็นส่วนประกอบของหัว เพื่อที่จะนำตำแหน่งของมือไปใช้เป็น Local Position ทำให้หัวของผู้เล่นเป็นจุดอ้างอิงของมือทั้งสอง



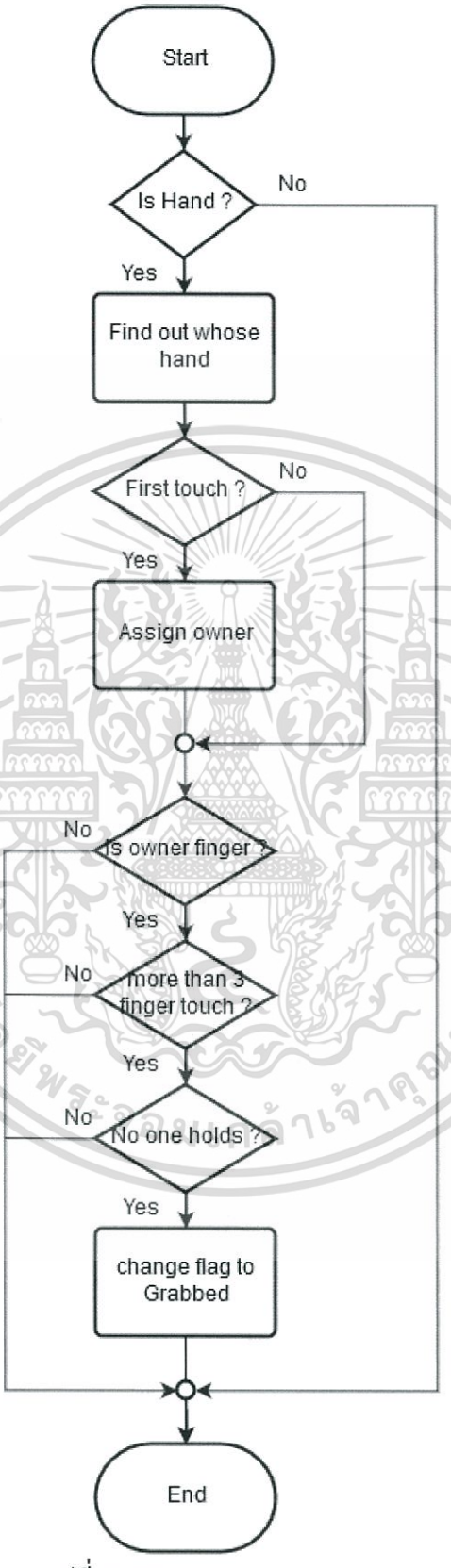
รูปที่ 3.16 การอ้างอิงตำแหน่งของหัวและมือของแต่ละโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 Interaction

เมื่อทราบถึงตำแหน่งของมือในตำแหน่งโลกจริงแล้วนั้น เราได้ทำการสร้าง Interaction Behavior Script สำหรับตรวจสอบการชนกันระหว่างปลายนิ้วกับวัตถุที่สามารถหยิบจับได้ เพื่อสร้างเป็นเงื่อนไขการหยิบจับ ตาม Flowchart รูปที่ 3.17 ถ้าเข้าเงื่อนไขการหยิบจับแล้ว Position ของ Object ที่ถูกหยิบนั้นจะไปอยู่จุดกึ่งกลางระหว่างนิ้วทุกนิ้วที่ทำการหยิบจับ และมี Rotation เป็นค่าเดียวกันจุดศูนย์กลางมือ หรือฝ่ามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 Interaction Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบเกมกระดาน The Lost Travelers

3.4.1 รายละเอียดของเกม

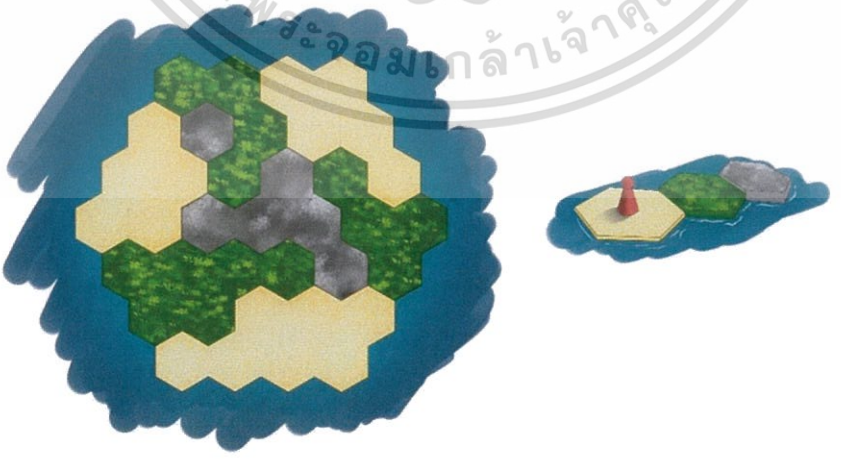
The Lost Traveler เป็นเกมกระดานแนวเอาชีวิตรอดที่ใช้ผู้เล่นทั้งหมด 2 คน ผู้เล่นทั้งสองจะได้รับบทเป็นนักท่องเที่ยวที่ประสบอุบัติเหตุทางทะเลแล้วลอยมาติดยังเกาะร้างแห่งหนึ่ง ในขณะที่มรสุมพัดเข้าสู่เกาะทำให้เกาะค่อย ๆ จมลงในน้ำ ผู้เล่นทั้งสองจึงต้องทำการสร้างแพเพื่อออกไปจากเกาะแห่งนี้ พร้อมกับต้องหาอาหารและน้ำมาดำรงชีวิต รวมถึงต้องหลบหนีสัตว์ป่าที่เริ่มตื่นจากการจำศีล แล้วออกมาไล่ล่าเหยื่ออีกด้วย

3.4.2 เป้าหมายของเกม

รวบรวมสิ่งของเพื่อสร้างแพ กักตุนอาหารแล้วหนีออกไปจากเกาะก่อนที่ทั้งเกาะจะจมลงในทะเล โดยสิ่งที่ผู้เล่นจะต้องรวบรวมเพื่อสร้างแพนั้นประกอบไปด้วย ท่อนไม้ 10 ท่อน เถาวัลย์ 5 เส้น เนื้อสัตว์ 5 ชิ้น เมื่อรวบรวมเสร็จสิ้นแล้ว ผู้เล่นคนสุดท้ายที่รวบรวมไอเทมได้จะได้รับไอเทม "แพ" สำหรับใช้เพื่อสร้างแพเมื่ออยู่ในพื้นที่ริมน้ำ โดยผู้เล่นที่ใช้ไอเทม “แพ” สำเร็จจะเป็นผู้รอดชีวิตทันที ส่วนผู้เล่นอีกคนจะต้องเดินมาให้ถึงช่องที่ทำการสร้างแพ จึงจะเป็นผู้รอดชีวิต

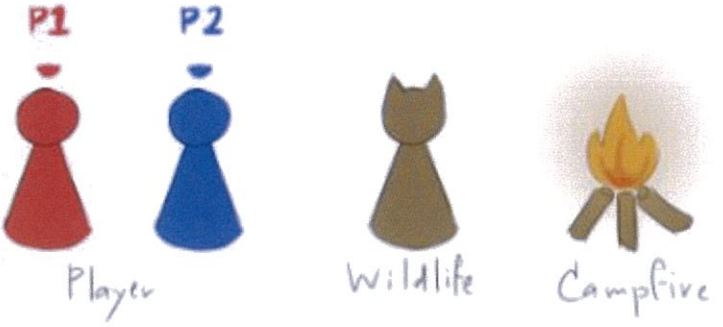
3.4.3 แผนที่ของเกม

แผนที่ของเกมจะถูกแบ่งเป็นช่อง 6 เหลี่ยม จำนวน 37 ช่อง โดยแต่ละช่องจะมีลักษณะและความสูงอยู่ 3 ระดับคือ ชายหาด ป่า และภูเขา โดยลักษณะทั้งสามนั้นจะมีความแตกต่างกันในด้านของไอเทมที่หาได้ และลำดับการจมลงทะเล



รูปที่ 3.18 แผนที่ ระดับความสูงของช่องในภูมิประเทศต่าง ๆ และการวางตัวละครบนช่อง

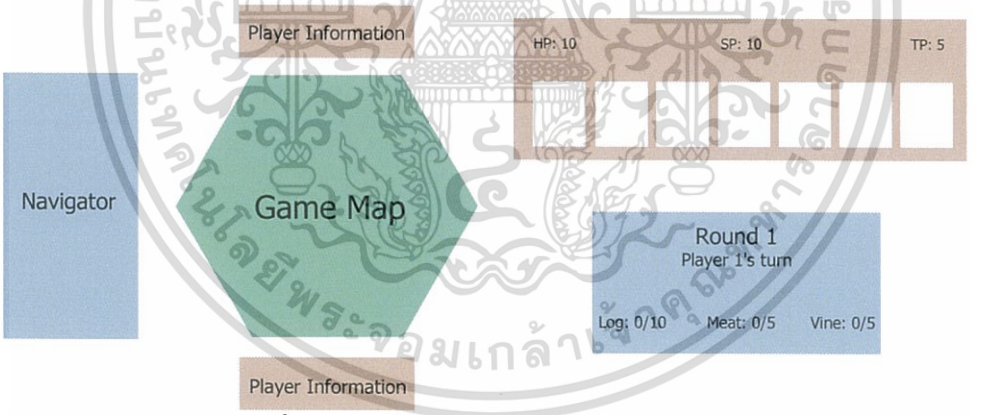
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 วัตถุต่าง ๆ ที่วางบนแผนที่

3.4.4 สภาพแวดล้อมของเกม

สภาพแวดล้อมของเกม The Lost Travelers จะให้ผู้เล่นทั้งสองนั่งอยู่ฝั่งตรงข้ามกัน โดยมีแผนที่เกมอยู่ตรงกลางระหว่างผู้เล่นทั้งสอง โดยด้านหน้าของผู้เล่นแต่ละคนจะมีข้อมูลตัวละคร และการ์ดไอเทมที่มีอยู่ นอกจากนี้ ด้านข้างของผู้เล่นยังมี หน้าต่าง Navigator สำหรับแสดงสถานะปัจจุบันของเกมอีกด้วย



รูปที่ 3.20 การจัดวางสภาพแวดล้อมของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 กฎ กติกา และวิธีการเล่น

3.4.5.1 ลำดับการเล่น

เป็นการเล่นเกมในแต่ละรอบ โดยจะมีลำดับการเล่นในแต่ละรอบเกมดังนี้

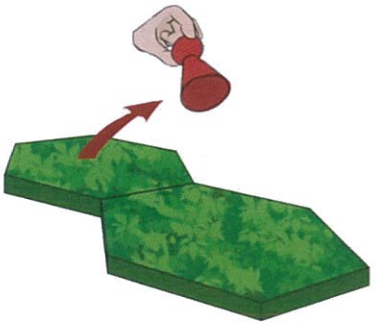
1) Player1's Turn เป็นตาเดินของผู้เล่นคนที่ 1
2) Player2's Turn เป็นตาเดินของผู้เล่นคนที่ 2
3) Wildlife Move สัตว์ป่าที่ปรากฏตัวบนแผนที่แล้วจะสุ่มเดินเป็นระยะทาง 1 ช่อง
4) Consumption Resolve ช่วงตรวจสอบค่า SP และ TP ของผู้เล่นแต่ละคน เพื่อลงโทษผู้เล่นตามเงื่อนไข หากมีค่าใดค่าหนึ่งหมดไปจากนั้นลดค่า SP และ TP ของผู้เล่นลงตามที่กำหนด
5) Flooding เป็นการจมพื้นที่ของเกาะลงไปในทะเล โดยจะสุ่มจมจากพื้นที่ที่มีระดับต่ำสุดจำนวน 4 ช่อง ในแต่ละรอบเกม หากผู้เล่น หรือสัตว์ป่าอยู่บนช่องที่ทำการจม จะถูกนำออกจากเกม หากแพอยู่บนช่องที่ทำการจม จะถูกย้ายไปยังช่องข้างเคียงแบบสุ่ม

3.4.5.2 ตาเดินของผู้เล่น

ในตาเดินของผู้เล่นจะมีลำดับดังนี้

1) ใช้ไอเทม - ผู้เล่นสามารถใช้ไอเทมที่มีอยู่เพื่อรับผลของไอเทมนั้น ๆ ได้ โดยการแตะที่ไอเทมที่ต้องการ ไอเทมที่ถูกแตะจะลอยขึ้นมาเพื่อรอการยืนยันเป็นเวลา 3 วินาที เมื่อแตะอีกครั้งจะเป็นการใช้ไอเทม หากผู้เล่นทำการเดินแล้วจะไม่สามารถใช้ไอเทมได้อีก
2) เดิน - เมื่อใช้ไอเทมที่ต้องการจนครบแล้ว ผู้เล่นจะต้องหยิบตัวหมากเพื่อเลือกเดินไปยังช่องที่อยู่รอบ ๆ โดยจะสามารถเดินได้รอบเกมละ 1 ช่อง เมื่อเดินเสร็จแล้วผู้เล่นจะต้องทอยลูกเต๋า หากได้แต้ม 1-2 ผู้เล่นจะสูญเสีย HP 1 หน่วย หากได้แต้ม 3-5 ผู้เล่นจะได้รับไอเทมจำนวน 2 ใบ และหากได้ 6 แต้ม ผู้เล่นจะได้รับไอเทมจำนวน 3 ใบ โดยผู้เล่นแต่ละคนจะมีไอเทมบนมือได้ไม่เกิน 7 ชิ้น หากช่องไอเทมไม่เพียงพอจะไม่ได้รับไอเทมเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

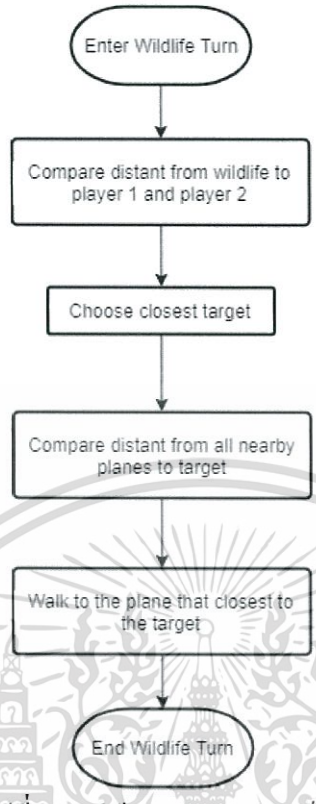


รูปที่ 3.21 ตัวอย่างการเดินจากช่องหนึ่งไปยังอีกช่องหนึ่ง

3.4.5.3 สัตว์ป่า

ในการเดินแต่ละครั้งของผู้เล่น ผู้เล่นจะมีโอกาสได้เจอกับหมีป่าที่นอนจำศีลอยู่ในเกาะแห่งนี้ โดยจะมีหมีป่าอยู่บนเกาะเป็นจำนวน 1 ตัว เมื่อผู้เล่นเดินไปยังช่องที่มีสัตว์ป่าอยู่ครั้งแรก จะเป็นการเปิดตำแหน่งของสัตว์ป่าบนแผนที่เกาะ ถัดมาสัตว์ป่าจะตื่นขึ้นในรอบถัดไป หลังจากนั้นจะเริ่มออกล่าในช่วง “ตาเดินของสัตว์ป่า” ของทุก ๆ รอบเกม โดยที่สัตว์ป่าจะสามารถเดินได้รอบเกมละ 1 ช่อง โดยจะเดินตามผู้เล่นที่อยู่ใกล้มากที่สุด และหากสัตว์ป่าเดินไปยังช่องที่มีผู้เล่นอยู่ ผู้เล่นจะเสียชีวิตและพ่ายแพ้ในเกมนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 วิธีการเดินของสัตว์ป่า

3.4.5.4 ข้อมูลผู้เล่น

ตัวละครที่ใช้แทนตัวผู้เล่นแต่ละคน โดยจะมีค่าสถานะดังนี้

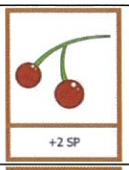
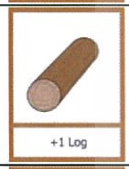
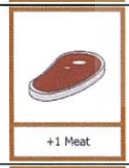
1) Health Point (HP) – หรือค่าพลังชีวิตของผู้เล่น ซึ่งผู้เล่นแต่ละคนจะมีค่า HP เริ่มต้นที่ 10 หน่วย โดยหากค่า HP ลดลงถึง 0 หน่วย ผู้เล่นจะเสียชีวิตและพ่ายแพ้ในเกมนี้

2) Starvation Point (SP) – หรือค่าความหิวของผู้เล่น ซึ่งผู้เล่นแต่ละคนจะมีค่า SP เริ่มต้นที่ 10 หน่วย และจะลดลงทีละ 3 หน่วยในช่วง “คำนวนสถานะผู้เล่น” โดยหากค่า SP ลดลงถึง 0 หน่วย ในรอบถัดไป ผู้เล่นจะได้รับไอเทมจากการเดินและการค้นหาน้อยลง 1 ชิ้น

3) Thirst Point (TP) - หรือค่าความกระหายของผู้เล่น ซึ่งผู้เล่นแต่ละคนจะมีค่า TP เริ่มต้นที่ 5 หน่วย และจะลดลงทีละ 2 หน่วยในช่วง “คำนวนสถานะผู้เล่น” โดยหากค่า TP ลดลงถึง 0 หน่วย ผู้เล่นจะสูญเสีย HP 3 หน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.1 รายละเอียดของการ์ดไอเทม

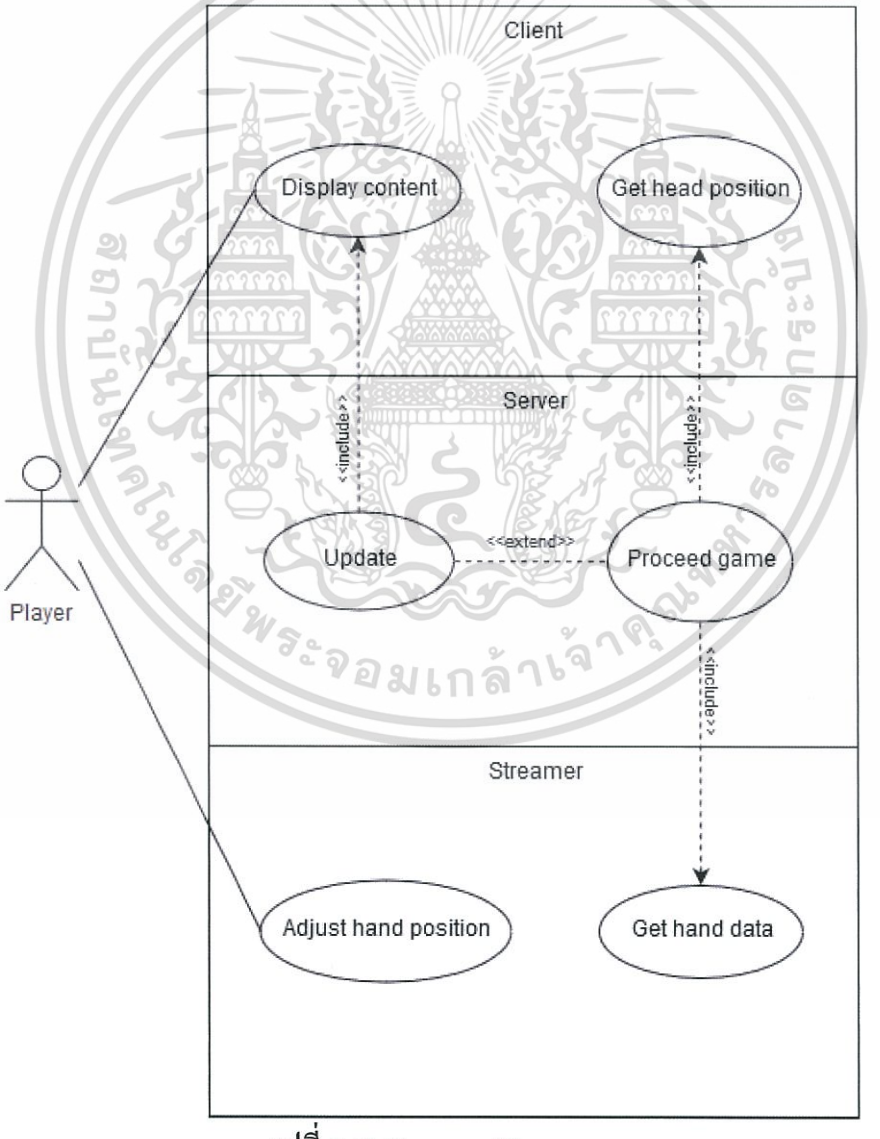
ภาพไอเทม	ชื่อไอเทม	แหล่งที่หาได้	ระดับความหายาก	ผลของไอเทม
	เบอร์รี่	ป่า	ง่าย	ฟื้นฟู SP 2 หน่วย
		ภูเขา	ยาก	
	มะพร้าว	ชายหาด	ปานกลาง	ฟื้นฟู SP 1 หน่วย TP 2 หน่วย
	ปลา	ชายหาด	ง่าย	ฟื้นฟู SP 2 หน่วย
		ป่า	ปานกลาง	
	ปู	ชายหาด	ง่าย	ฟื้นฟู SP 1 หน่วย
	น้ำจืด	ป่า	ปานกลาง	ฟื้นฟู TP 2 หน่วย
		ภูเขา	ยาก	
	สมุนไพร	ป่า	ง่าย	ฟื้นฟู HP 2 หน่วย
		ภูเขา	ปานกลาง	
	ท่อนไม้	ป่า	ปานกลาง	เพิ่มจำนวนไม้ให้กับเป้าหมาย 1 ชิ้น
		ภูเขา	ปานกลาง	
	เนื้อสัตว์	ป่า	ปานกลาง	เพิ่มจำนวนเนื้อสัตว์ให้กับเป้าหมาย 1 ชิ้น
		ภูเขา	ยาก	
	เถาวัลย์	ป่า	ปานกลาง	เพิ่มจำนวนเถาวัลย์ให้กับเป้าหมาย 1 ชิ้น
		ภูเขา	ยาก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Use case diagram

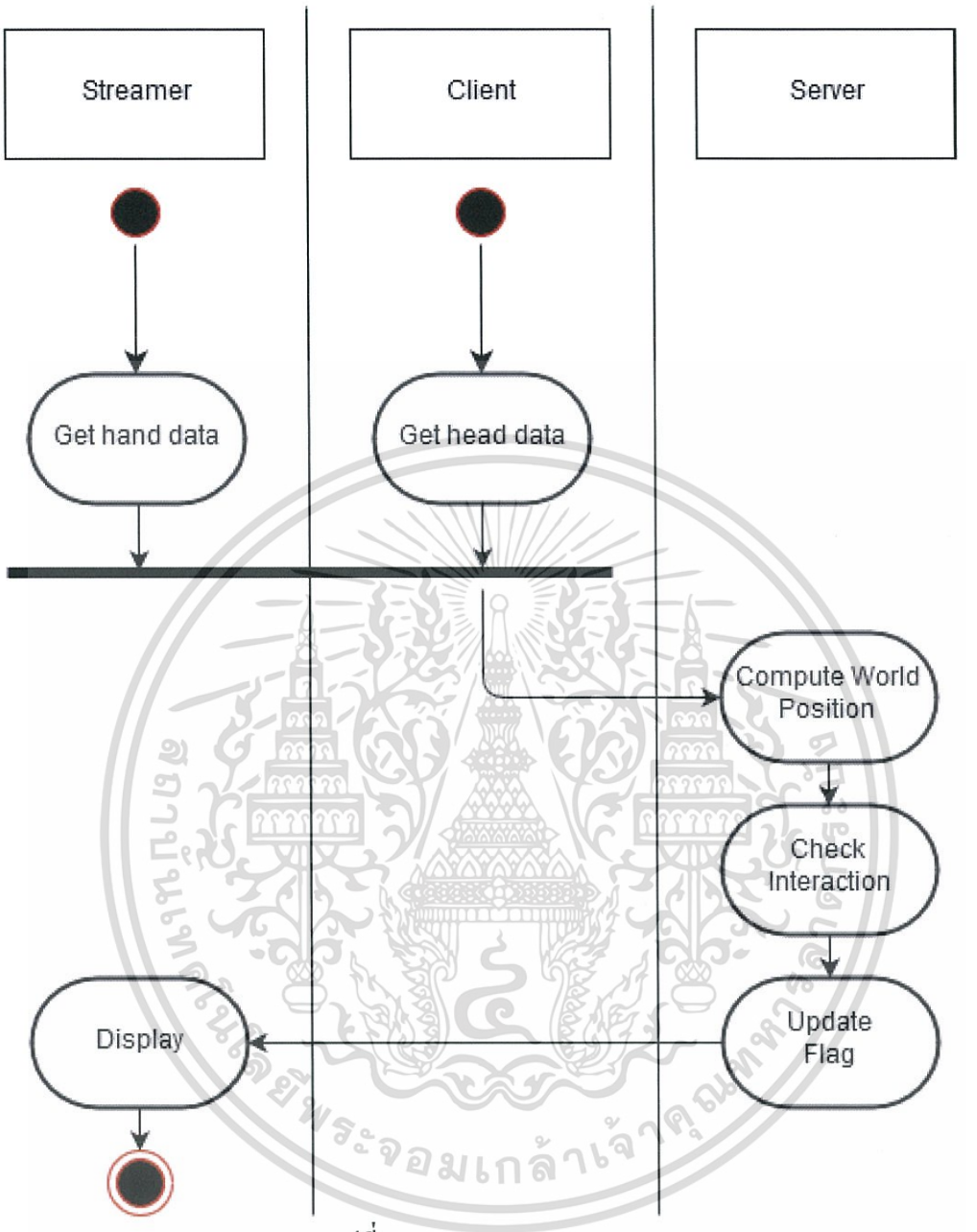
ในส่วนของการใช้งานของผู้ใช้นั้นจะแบ่งเป็นส่วนต่าง ๆ ตามส่วนประกอบของระบบ ดังนี้

1) Client ผู้เล่นสามารถดูข้อมูลต่าง ๆ รวมถึงการแสดงผลของเกม เช่น Item โดยข้อมูลจาก Server และส่งข้อมูลตำแหน่งหัวให้กับ Server
2) Server จะรับข้อมูลตำแหน่งมือจาก Streamer และข้อมูลตำแหน่งหัวของผู้เล่นจาก Client เพื่อไปประมวลผลแล้วส่งข้อมูลการแสดงผลทุกอย่างไปให้ Client
3) Streamer รับท่าทางของมือผู้เล่นไปแปลเป็นคำสั่งแล้วส่งไปดำเนินเกมต่อใน Game Server และสามารถปรับตำแหน่งของมือผู้เล่นในเกมเพื่อให้ตรงกับมือของผู้เล่นในความเป็นจริงได้



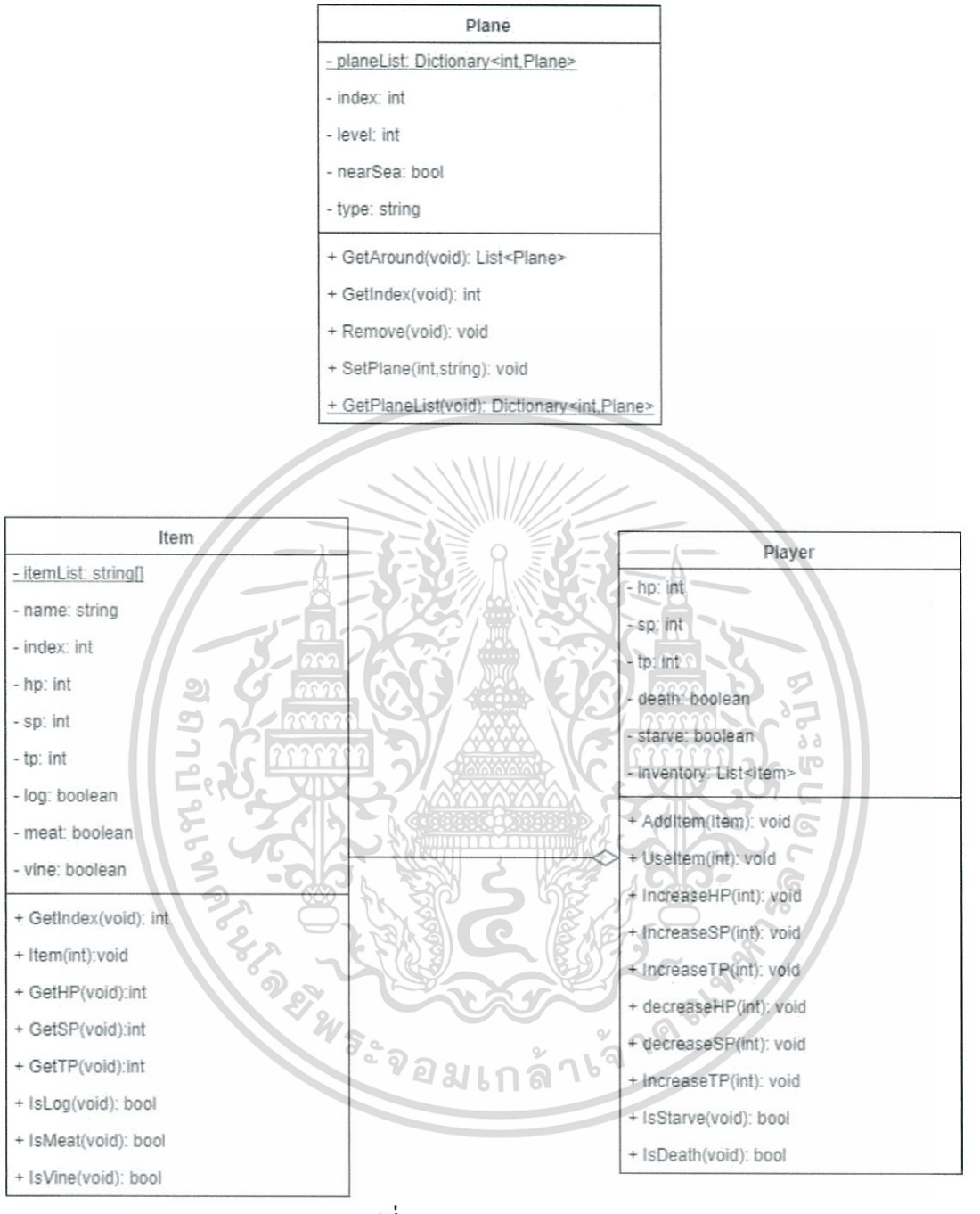
รูปที่ 3.23 Use case Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 Activity Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 Class Diagram

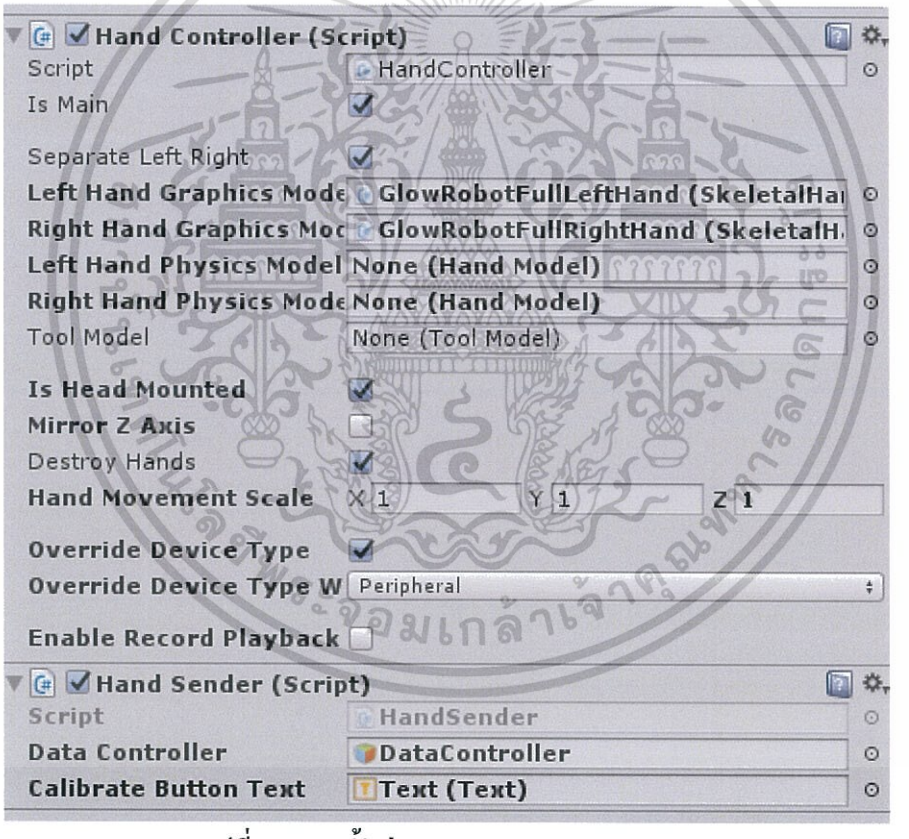
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 โปรแกรมสำหรับส่งข้อมูล Leap Motion

การทำงานหลัก ๆ ของโปรแกรมคือการอ่านข้อมูล Leap Motion แล้วส่งไปให้ Server ประมวลผล แต่เนื่องด้วยภาพที่ส่งมาจาก Leap Motion เป็นภาพ 2 มิติ เพราะฉะนั้นแหละเราจะใช้ Hand Controller Script ที่รวมอยู่ใน Leap Motion SDK มาแสดงผลใน Unity เพื่อที่จะดึงตำแหน่งของมือในแกน x, y, z มาใช้ โดยเราจะตั้งค่า Script ไว้ดังรูปที่ 4.1



รูปที่ 4.1 การตั้งค่า Hand Controller Script

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากมีโมเดลมือปรากฏใน Unity แล้วนั้นเราจะสังเกตเห็นได้ว่าโมเดลมือนั้นมี hierarchy ที่ประกอบไปด้วยข้อต่าง ๆ ตามนิ้วมือ ดังรูปที่ 4.2 เราจะใช้ Position นิ้วล่ะ 3 ข้อต่อ และฝ่ามือ ส่งไปให้ Server เพื่อประมวลต่อไปโดย เราจะเอาตำแหน่งของมือมาแปลงเป็น String แล้วคั่นด้วยเครื่องหมาย "," ระหว่างค่า Float ของ Position แต่ล่ะส่วนที่จะส่งไป

โปรแกรมที่ 4.1 ส่วนหนึ่งของฟังก์ชันสำหรับข้อความตำแหน่งมือ

```
strMessage = "hand,";
if (transform.Find("GlowRobotFullLeftHand(Clone)") != null) {
      Transform leftHand =
transform.Find("GlowRobotFullLeftHand(Clone)");
      Transform palm = leftHand.transform.GetChild(2);
      Transform thumb = leftHand.transform.GetChild(5);
      Transform index = leftHand.transform.GetChild(0);
      Transform middle = leftHand.transform.GetChild(1);
      Transform ring = leftHand.transform.GetChild(4);
      Transform pinky = leftHand.transform.GetChild(3);
      Vector3 handPosition = palm.position + offSet;
      strMessage += "1," + handPosition.ToString() + "," +
palm.rotation.ToString() + ",";

      Vector3[] bone = new Vector3[15];
      for (int i = 0; i < 3; i++) {
            bone[0 + i] = thumb.transform.GetChild(i).position
      + offSet;
            bone[3 + i] = index.transform.GetChild(i).position
      + offSet;
            bone[6 + i] = middle.transform.GetChild(i).position
      + offSet;
            bone[9 + i] = ring.transform.GetChild(i).position +
      offSet;
            bone[12 + i] = pinky.transform.GetChild(i).position
      + offSet;
      }
      for (int i = 0; i < 15; i++) {
                  strMessage += bone[i].ToString() + ",";
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 hierarchy ของโมเดลมือ

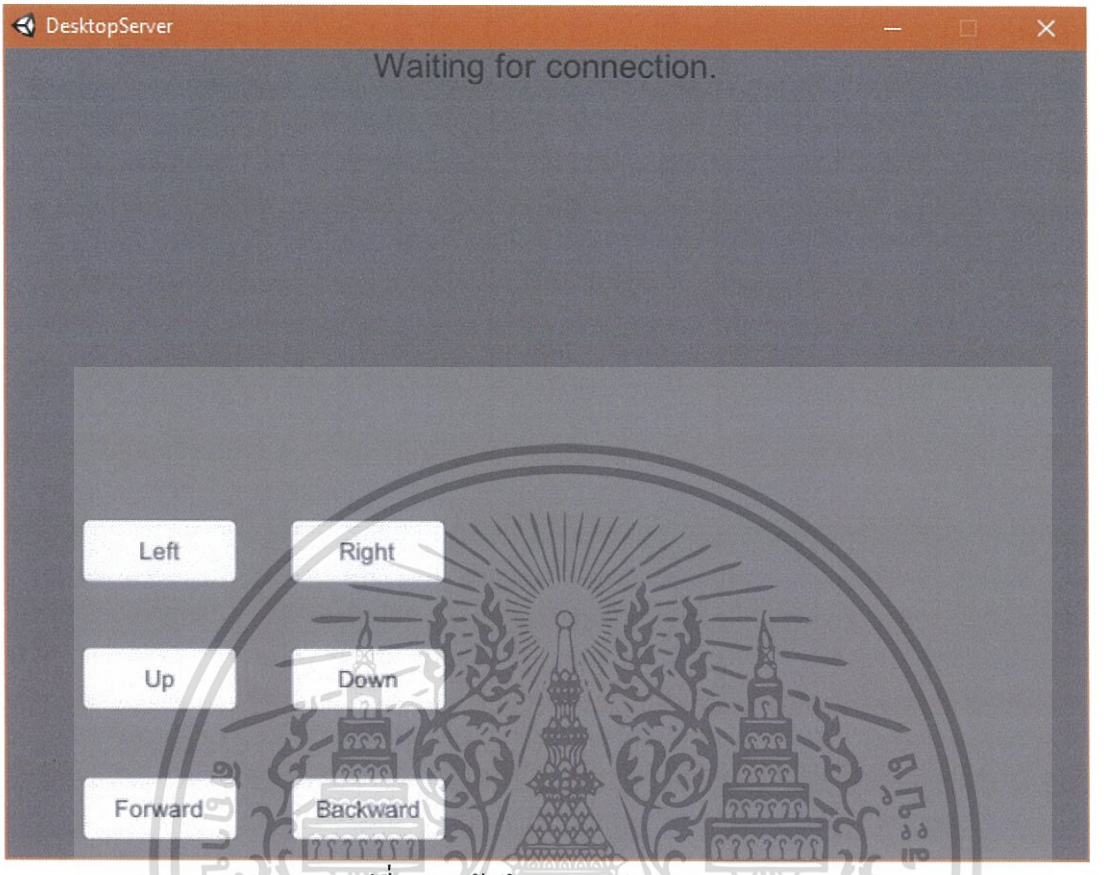
การส่งข้อมูลไปยัง Server นั้นเราจะใช้ UdpClient เป็นตัวช่วยในการส่ง แต่ก่อนหน้านั้นต้องแปลง String เป็น Byte[] แล้วใช้ Method Send ส่งไปยัง gameServer ที่เป็นตัวแปร IPEndPoint ของ Server

โปรแกรมที่ 4.2 ฟังก์ชัน SendData

```
public void SendData(string msg) {
      if(isConnected) {
            byte[] send_data = Encoding.UTF8.GetBytes(msg);
            terminal.Send(send_data, send_data.Length,
      gameServer );
      }
}
```

สำหรับ UI จะมีปุ่ม ที่เอาไว้ปรับ Position ในแกนต่าง ๆ ของมือเพื่อปรับตำแหน่งของมือที่คลาดเคลื่อนให้มีความแม่นยำ โดยการกดปุ่ม 1 ครั้งจะเป็นการบวกลบค่า Offset ในแกนนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าต่าง UI ของ Streamer

4.2 แอปพลิเคชันบนสมาร์ทโฟน

การแสดงผลบน Smartphone นั้นจะรับข้อความมาจาก Server โดยเมื่อข้อความมาถึงจะใช้ UdpClient เข้าไปอ่านข้อความใน Port 1998 แล้วแยกประเภทข้อความด้วย DataController Script โดยทั้งหมดนั้นจะถูกจัดการภายในฟังก์ชัน ReceiveData

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 4.3 ฟังก์ชัน ReceiveData

```
void ReceiveData() {
     print("Waiting for connection.");
     IPEndPoint anyIP = new IPEndPoint(IPAddress.Any, 0);
     while (true) {
          byte[] data = terminal.Receive(ref anyIP);
          string msg = Encoding.UTF8.GetString(data);
          text1.text = msg;
          if (msg.StartsWith ("client")) {
               ipEndPoint = anyIP.Address.ToString ();
               gameServer = new IPEndPoint
          (IPAddress.Parse (ipEndPoint), sendingPort);
               isConnected = true;
               status.text = "Connected to " +
          ipEndPoint;
               SendData ("clientok");
               headSender.InputMessage ("connected");
          } else if (msg.StartsWith ("hand")) {
               handReciever.InputMessage (msg);
          } else if (msg.StartsWith ("display")){
               displayGenerator.InputMessage (msg);
          } else if (msg.StartsWith ("navigator")) {
               navigator.InputMessage (msg);
          }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 4.3 ฟังก์ชัน ReceiveData (ต่อ)

```
            else if (msg.StartsWith ("playerinfo,p1")) {
                player1.InputMessage (msg);
            } else if (msg.StartsWith ("playerinfo,p2")) {
                player2.InputMessage (msg);
            } else if (msg.StartsWith ("playerinfo,setup"))
        {
                player1.Reset ();
                player2.Reset ();
            } else if (msg.StartsWith("disconnect")) {
                headSender.InputMessage("disconnect");
                handReciever.InputMessage("disconnect");
                displayGenerator.DestroyAllObject ();
                isConnected = false;
                status.text = "Waiting.";
            }
        }
}
```

ข้อความจะถูกส่งไปยัง Script ต่าง ๆ เพื่อจัดการกับข้อความที่ได้รับมา โดย HeadSender script จะจัดการเรื่องการส่งตำแหน่งของกล้อง HandReceiver script จัดการเรื่องการแปลงข้อความตำแหน่งมือให้เป็นข้อมูลการแสดงผล DisplayGenerator script จัดการเรื่องการแสดงผล Object ต่าง ๆ Navigator script จัดการการแสดงผลบนหน้าต่างดำเนินเกม และ Player script จัดการการแสดงผลข้อมูลสถานะผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 DisplayGenerator

ข้อความจะถูกเอาไปเช็คกับเงื่อนไขว่า ถ้าเป็น Object ที่มีอยู่แล้วก็จะดำเนินคำสั่ง “move”, “active”, “inactive” ตามข้อความด้วยการ Set Position แต่ถ้า Object นั้นเป็น Object ที่ยังไม่ถูกสร้างขึ้น จะถูกดำเนินการใน InitObject ก่อน โดยจะเอาชื่อไปเทียบกับแล้ว Instantiate Object นั้นจาก Prefab หรือ Asset ใน Resource ของ Unity ที่ได้เตรียมไว้

โปรแกรมที่ 4.4 ฟังก์ชัน InitObject

```
void InitObject(string objName, Vector3 pos, Quaternion
rot, bool active) {
      GameObject temp = null;
      if (objName.StartsWith("grass")) {
            temp = Instantiate(grassBlock, transform);
      } else if (objName.StartsWith("rock")) {
            temp = Instantiate(rockBlock, transform);
      } else if (objName.StartsWith("sand")) {
            temp = Instantiate(sandBlock, transform);
      } else if (objName.StartsWith("P1Char")) {
            temp = Instantiate(player1, transform);
      } else if (objName.StartsWith("P2Char")) {
            temp = Instantiate(player2, transform);
      } else if (objName.StartsWith("dice"))          {
            temp = Instantiate(dice, transform);
      }
      if (temp != null) {
            temp.name = objName;
            temp.transform.position = pos;
            temp.transform.rotation = rot;
            objList.Add (temp.name, temp.transform);
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 HandReciever

การแสดงผลมือนั้นจะถูกจัดการในฟังก์ชัน UpdateHand ข้อความจะถูก Split ออกโดย "," แล้ว float.Parse แปลง string ให้เป็น float แล้วไป Set Position Rotation ส่วนต่าง ๆ ของมือ

โปรแกรมที่ 4.5 ส่วนหนึ่งของฟังก์ชัน UpdateHand

```
leftHandExists = true;
if (currentLeftHand == null) {
      currentLeftHand = Instantiate(leftHandPrefab,
this.transform);
}
currentLeftHand.transform.GetChild(0).localPosition = new
Vector3(float.Parse(sArray[2]), float.Parse(sArray[3]),
float.Parse(sArray[4]));
currentLeftHand.transform.GetChild(0).localRotation = new
Quaternion(float.Parse(sArray[5]), float.Parse(sArray[6]),
float.Parse(sArray[7]), float.Parse(sArray[8]));

for (int i = 1; i < currentLeftHand.transform.childCount; i++)
{
      currentLeftHand.transform.GetChild(i).localPosition = new
Vector3(float.Parse(sArray[6 + 3 * i]), float.Parse(sArray[7 +
3 * i]), float.Parse(sArray[8 + 3 * i]));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 HeadSender

Script นี้จะทำหน้าที่ส่งตำแหน่งหัวหรือตำแหน่งของ AR Camera ไปให้กับ Server เพื่อประมวลผลตำแหน่งจริง ๆ ใน World Position โดยหลังจาก Device เสร็จสิ้นการเชื่อมต่อกับ Server แล้วนั้น จะส่งข้อความตำแหน่งผ่าน SendData ข้อความที่จะส่งนั้นจะถูกเทียบก่อนส่งว่าซ้ำกับข้อความครั้งล่าสุดที่เคยส่งไปหรือไม่ เพื่อลดปริมาณ Traffic ของ data ที่ส่ง

โปรแกรมที่ 4.6 ฟังก์ชัน SendData

```
void SendData() {
      while(true) {
            if (isConnected) {
                  msg = "head,";
                  if (isTracked) {
                        msg += transform.position.ToString() +
                  "," + transform.rotation.ToString();
                        msg = msg.Replace("(", "");
                        msg = msg.Replace(")", "");
                        msg = msg.Replace(" ", "");
                  }
                  else {
                        msg += "nothing";
                  }
                  if (msg != prevMsg) {
                        dataCtrl.SendData(msg);
                        prevMsg = msg;
                  }
            }
      }
}
```

4.2.4 Navigator

ข้อความที่ขึ้นต้นด้วย “navigator” จะถูกจัดการใน script นี้ โดย navigator จะแสดงข้อมูลรอบที่เล่น ตาเดินของใคร และข้อมูลทรัพยากรในชนะเกม โดยข้อมูลที่ได้หลักจาก Decode ให้เป็น string แล้วนั้นจะถูกเอาไปแสดงโดยตรงในหน้าต่างดำเนินเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 Player

หน้าต่างสถานะตัวละคร และการแสดงผลการ์ดไอเทมต่าง ๆ จะถูกจัดการใน script นี้ เมื่อข้อความที่ขึ้นต้นด้วย “playerinfo” ถ้าข้อความตามด้วย “hp”, “sp”, “tp” ให้แสดง Value ของมันโดยการนำ string ไปใส่ใน text ได้เลย แต่ถ้าหากมีการเปลี่ยนแปลงการแสดงผลของการ์ดไอเทมข้อมูลจะถูกส่งมาเป็นชุด 7 ค่าโดย เลขต่าง ๆ หมายถึง index ของภาพที่ได้เตรียมไว้และ “n” หมายถึง ช่องว่างจะใช้ภาพที่เป็น default

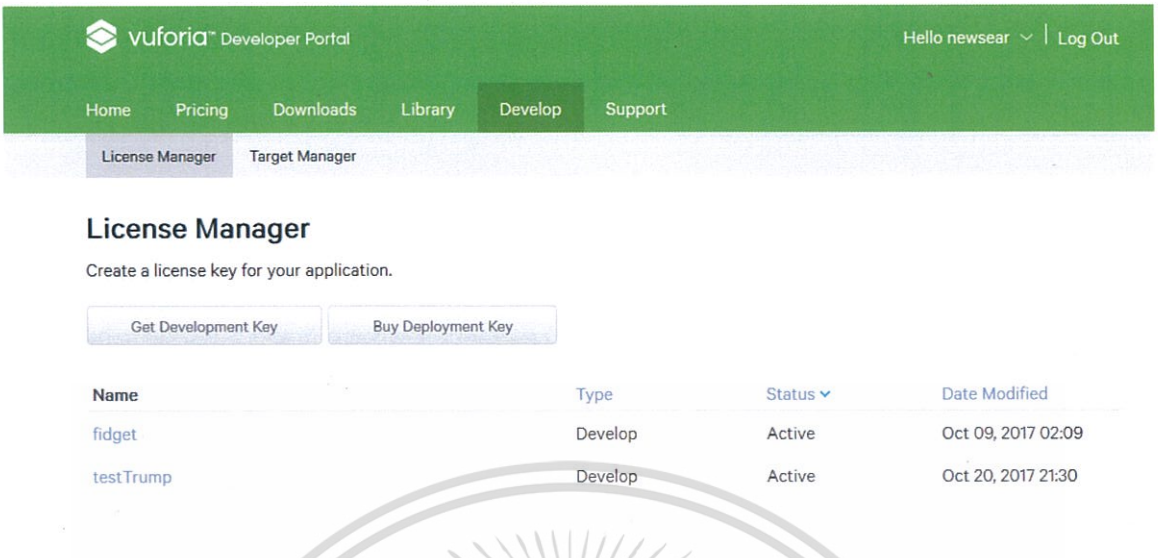
โปรแกรมที่ 4.7 ฟังก์ชันการจัดการการแสดงผลการ์ดไอเทม

```
for(int i = 0; i < sArray.Length-3; i++) {
      MeshRenderer card =
      itemSlot.GetChild(i).GetComponent<MeshRenderer>();
      if (sArray [i + 3] == "n") {
            card.material = defaultMat;
      } else {
            card.material = itemMats [int.Parse (sArray [i +
      3])];
      }
}
```

4.2.6 Vuforia

การแสดงผลทุกอย่างจะอยู่ในรูปแบบของ AR โดยมี Object ที่จะแสดงผลจะเป็น child ของ Image target สำหรับการจัดการนั้น ต้อง Import Asset Database ของ Image target มาจาก Vuforia เสียก่อน ขั้นแรกเข้าไปที่จัดการขอ License key เพื่อให้สามารถใช้งาน AR ของ Vuforia ได้ ที่ Developer Portal

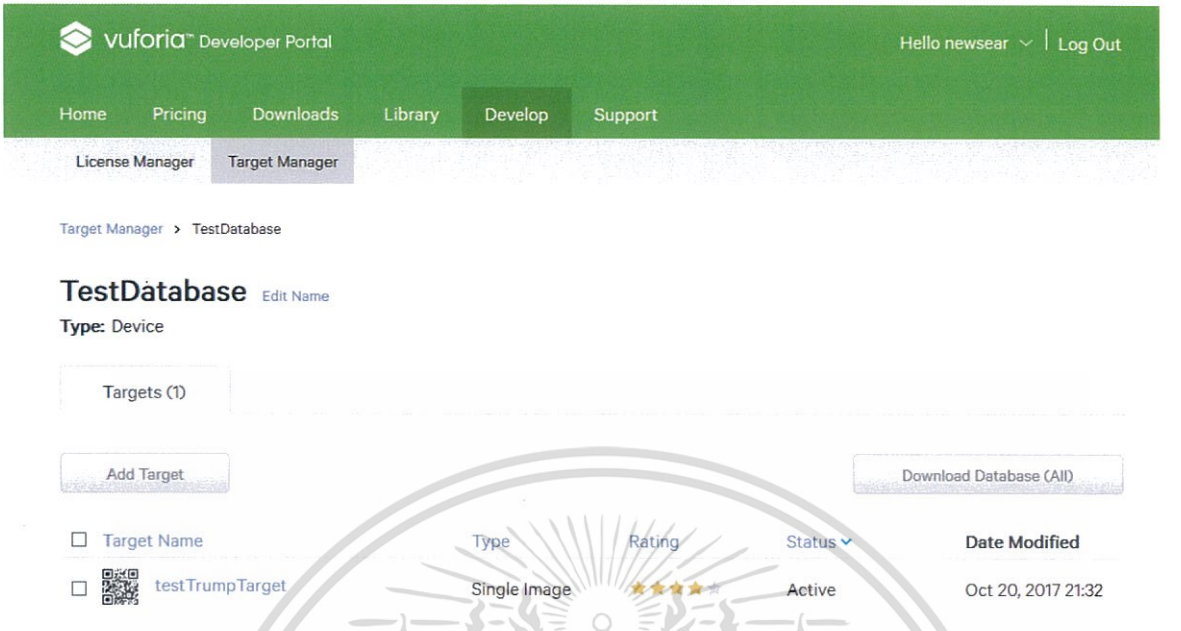
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Vuforia License Manager

เมื่อเราได้ License key มาแล้วให้ไปสร้าง Image target ที่เป็นตัวสำหรับการแสดงผล เมื่อกล้องจับภาพ target นี้ได้ จะแสดง Object ที่เราเตรียมไว้ขึ้นมา โดยในที่นี้เราใช้เป็นภาพของ QR code เพราะมีความ unique กับสภาพแวดล้อมรอบข้าง ทำให้จับภาพได้ง่าย หลังจากนั้นให้โหลดไฟล์มาในรูปแบบ asset ของ Unity เพื่อ Import เข้าไปใน Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 Vuforia Target Manager

Object ที่เป็น child ของ Image target จะถูกแสดงขึ้นเมื่อกล้องจับภาพหรือสิ่งของที่เราได้ตั้งไว้ตาม database ใน Vuforia ดังรูปที่ 4.6

รูปที่ 4.6 Vuforia AR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 Server

4.3.1 UDP Service

จะเป็นสคริปต์สำหรับควบคุมการรับส่งข้อมูลทั้งหมด โดยจะมีช่องทางรับข้อมูลทางพอร์ท 1999 และส่งข้อมูลทางพอร์ท 1998 โดยเมื่อผู้ใช้ทำการกรอก IP Address ของแต่ละอุปกรณ์เรียบร้อยแล้ว เมื่อกดปุ่มเชื่อมต่อ Server จะทำการบันทึก IP Address ทั้งหมด และส่งข้อความเพื่อเริ่มการเชื่อมต่อ จากนั้น หากอุปกรณ์ตอบกลับมาก็จะเริ่มการทำงานของ Server ทันที

รูปที่ 4.4 หน้าต่าง UI ของ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 4.8 ฟังก์ชัน ButtonPressed

```
public void ButtonPressed() {
      InputField streamer1 =
GameObject.Find("Streamer1IP").GetComponent<InputField>();
      InputField client1 =
GameObject.Find("Client1IP").GetComponent<InputField>();
      InputField streamer2 =
GameObject.Find("Streamer2IP").GetComponent<InputField>();
      InputField client2 =
GameObject.Find("Client2IP").GetComponent<InputField>();
      streamerIP.Clear();
      clientIP.Clear();
      streamerIP.Add(new IPEndPoint(IPAddress.Parse(streamer1.text),
1998));
      clientIP.Add(new IPEndPoint(IPAddress.Parse(client1.text),
1998));
      streamerIP.Add(new IPEndPoint(IPAddress.Parse(streamer2.text),
1998));
      clientIP.Add(new IPEndPoint(IPAddress.Parse(client2.text),
1998));
      BroadcastData("streamer", streamerIP);
      BroadcastData("client", clientIP);
}
```

ในส่วนของฟังก์ชัน ReceiveData จะทำหน้าที่คอยรับข้อความทางพอร์ท 1999 แล้วนำไปดำเนินการต่อไป นอกจากนี้ยังมีหน้าที่ตรวจสอบเงื่อนไขการเริ่มทำงานของ Server อีกด้วย โดยเมื่ออุปกรณ์ต่าง ๆ ตอบกลับมายัง Server ครบแล้วจึงจะเริ่มการทำงานของ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 4.9 ฟังก์ชัน ReceiveData

```
void ReceiveData() {
      IPEndPoint sender_ip = new IPEndPoint(IPAddress.Any, 0);
      while (true) {
            byte[] receive_data = receiver.Receive(ref sender_ip);
            if (send from streamer 1) {
                  msg[0] = Encoding.UTF8.GetString(receive_data);
            }
            else if (send from client 1) {
                  msg[1] = Encoding.UTF8.GetString(receive_data);
            }
            else if (send from streamer 2) {
                  msg[2] = Encoding.UTF8.GetString(receive_data);
            }
            else if (send from client 2) {
                  msg[3] = Encoding.UTF8.GetString(receive_data);
            }
      }
}
```

โปรแกรมที่ 4.10 โค้ดเทียมสำหรับเช็คความพร้อม

```
if(!allReady) {
      if(receive ack from streamer 1) {
            streamer 1 ready;
      }
      if(receive ack from client 1) {
            client 1 ready;
      }
      if(receive ack from streamer 2) {
            streamer 2 ready;
      }
      if(receive ack from client 2) {
            client 2 ready;
      }
      if(all device is ready) {
            allReady = true;
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 DisplaySender

เป็นสคริปต์สำหรับส่งข้อมูลของวัตถุบนกระดานไปยัง Client โดยจะตรวจหา GameObject ทุกชิ้นที่มี Hierarchy อยู่ภายใต้ GameBoard และมี Tag เป็น GameComponent เมื่อเจอ GameObject ที่ตรงตามเงื่อนไข ก็จะทำการส่งข้อความ display ตามเงื่อนไขที่กำหนด

โปรแกรมที่ 4.11 ฟังก์ชัน ScanChild

```
void ScanChild() {
      Transform[] temp =
GetComponentsInChildren<Transform>(true);
      for(int i = 0; i < temp.Length; i++) {
            if (temp[i].tag == "GameComponent" && temp[i] !=
      transform) {
                  if (!children.Contains(temp[i].gameObject)) {
                        children.Add(temp[i].gameObject);
                  }
            }
      }
}
```

โปรแกรมที่ 4.12 โค้ดเทียมของฟังก์ชัน Update ใน DisplaySender

```
void Update() {
      if(allReady) {
            ScanChild();
            foreach(GameObject child in children) {
                  if(object has changed) {
                        if(object not active) {
                              sendingMsg = object active
                        }
                        else {
                              sendingMsg = object inactive
                        }
                  }
                  else if(object has moved) {
                        sendingMsg = object move to position
                  }
            }
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 InteractionBehavior

สคริปต์นี้จะถูกติดตั้งให้กับ GameObject ทุกชิ้นที่สามารถหยิบจับได้ ซึ่งจะต้องมี Collider และ Rigidbody ติดตั้งอยู่บน GameObject ด้วย โดยสคริปต์นี้จะทำการตรวจจับการชนของวัตถุอื่น และตรวจสอบเงื่อนไขการหยิบจับตามที่กำหนดไว้ เมื่อผ่านทุกเงื่อนไขก็จะทำการย้ายตำแหน่งของ GameObject ไปยังจุดกึ่งกลางของแต่ละ Object ที่ทำการจับวัตถุนี้เอาไว้

โปรแกรมที่ 4.13 โค้ดเทียมของฟังก์ชัน OnTriggerEnter

```
void OnTriggerEnter(Collider other) {
      if(other is hand) {
            fingerOwner = player who touch
            if(no owner) {
                  owner = fingerOwner;
            }
            if(owner == fingerOwner) {
                add to list
                if(finger in list more than 2) {
                      this object is child of hand
                        isGrabbed = true;
                }
            }
      }
}
```

4.3.4 Navigator

สคริปต์นี้จะทำงานเกี่ยวกับการดำเนินเกมทุกอย่าง ตั้งแต่การจัดกระดาน ลำดับการเล่นในแต่ละรอบ รวมถึงการสุ่มต่างๆที่เกิดขึ้นในเกมนี้ และส่งข้อความ navigator และ playerinfo ไปยัง Client เพื่อทำการแสดงข้อมูลให้แก่ผู้เล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 ผลการดำเนินงาน

1) เสร็จสิ้นการออกแบบรายละเอียดต่าง ๆ ของเกมกระดาน
2) เข้าใจวิธีการใช้งานโปรแกรม Unity 3D และสามารถ Import Vuforia มาใช้งานได้
3) PC สามารถส่งข้อมูล Leap Motion ที่เป็นตำแหน่งมือไปยังสมาร์ทโฟนได้
4) สมาร์ทโฟนสามารถรับภาพจากกล้องเพื่อมาประมวลผลและแสดงข้อมูล AR ออกมาได้
5) ระบบ AR สามารถทำงานร่วมกับ Leap Motion ได้
6) เสร็จสิ้นการสร้างระบบสำหรับการแสดงผล AR ร่วมกันให้แก่ผู้เล่นทั้งสอง
7) เสร็จสิ้นการสร้างการปฏิสัมพันธ์ในรูปแบบของการหยิบจับวัตถุเสมือน
8) เสร็จสิ้นการสร้างโมเดลและภาพสำหรับใช้ในเกมกระดาน
9) ดำเนินการสร้างเกมกระดานไปได้บางส่วน

5.2 ปัญหาและอุปสรรค

1) Leap Motion ปัจจุบันไม่สามารถต่อเข้ากับสมาร์ทโฟนได้โดยตรง จึงจำเป็นต้องใช้ PC เป็นตัวกลางในการส่งข้อมูล
2) Vuforia เวอร์ชั่น 6.5 ไม่สามารถใช้งานตัวเลือก Digital eye wear ได้ เนื่องจากเป็นข้อผิดพลาดจากทางผู้พัฒนา
3) ประสิทธิภาพการแสดงผลขึ้นอยู่กับสมาร์ทโฟนนั้น ๆ ถ้าสมาร์ทโฟนคุณภาพต่ำอาจจะทำให้มีการแสดงผลที่ล่าช้าบ้างในบางครั้ง
4) สำหรับระบบ AR และ Leap Motion แสงและฝุ่นเป็นปัจจัยหลักที่ทำให้ประสิทธิภาพการจับภาพแย่ลง
5) ไม่สามารถใช้งาน Interaction Module ของ Leap Motion ใน Server ได้ เนื่องจาก Game object ของมือใน Server ไม่ได้สร้างมาเพื่อรองรับ Leap Motion จึงต้องทำการสร้าง Interaction ในส่วนของการหยิบจับเอง
6) เนื่องจากระบบมีการรับส่งข้อความระหว่างโปรแกรมเยอะมาก ทำให้ต้องรอบคอบในการจัดการระบบมากขึ้น เพื่อไม่ให้ข้อความสูญหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) SDK ของ Leap Motion เวอร์ชั่นปัจจุบันไม่รองรับระบบปฏิบัติการ Linux ทำให้ไม่สามารถนำ Streamer ไปทำงานบน VM ได้ และหากใช้ระบบปฏิบัติการ Windows บน VM จะใช้ทรัพยากรเยอะมาก ทำให้ต้องใช้คอมพิวเตอร์จำนวน 2 เครื่องแทน

5.3 แนวทางในอนาคต

1) สร้าง Interaction ในด้านคำสั่งท่าทางต่าง ๆ ของมือ เพื่อนำมาใช้ในการเพิ่ม User Experience ในการเล่นเกมกระดาน
2) เพิ่มลูกเล่นของเกมดิจิตอล ให้กับเกมกระดาน เช่น ทัศนวิสัยที่ต่างกันของผู้เล่น เป็นต้น เพื่อเพิ่มความแปลกใหม่ให้กับเกมกระดาน
3) นำระบบมาประยุกต์ใช้กับสื่อมัลติมีเดียอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

TangoChen. 2016. “Coloreality.” [Online].

Available : https://github.com/TangoChen/Coloreality.

Matthew Hallberg. 2017. “Desktop Server Leap Motion.” [Online].

Available : https://github.com/MatthewHallberg/DesktopServerLeapMotion.

Leap Motion. 2017. “Interaction Engine.” [Online].

Available : https://github.com/leapmotion/UnityModules/wiki/Interaction-Engine.

Raffi Bedikian. 2015. “Augmented Reality Workspace.” [Online].

Available : http://blog.leapmotion.com/hood-leap-motion-hackathons-augmented-reality-workspace.

Vuforia. 2017. “Developer Portal.” [Online].

Available : https://library.vuforia.com.

Unity. 2017. “Documentation.” [Online].

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

Leap Motion. 2017. “Unity Modules.” [Online].

Available : https://github.com/leapmotion/UnityModules/wiki/Core.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้