

การพัฒนาเกมสำหรับการออกกำลังกาย

DEVELOPMENT OF GAME FOR THE EXERCISE



รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560

การพัฒนาเกมสำหรับการออกกำลังกาย  
DEVELOPMENT OF GAME FOR THE EXERCISE



รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาเกมเพื่อการออกกำลังกาย

DEVELOPMENT OF GAME FOR THE EXERCISE

ผู้จัดทำ

1. นายนภัทร พลอยสมบูรณ์ รหัสนักศึกษา 57010656

2. นายตีปกร สุภกิจญโญ รหัสนักศึกษา 57011352



อาจารย์ที่ปรึกษา

(รศ.ดร.เกียรติคุณ เจียรนัยระกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การพัฒนาเกมเพื่อการออกกำลังกาย

นายณภัทร พลอยสมบูรณ์ 57010656

นายสิปปกร ศุภภิญโญ 57011352

รศ.ดร.เกียรติคุณ เจียรนัยชนะกิจ อาจารย์ที่ปรึกษา

ปีการศึกษา 2560

## บทคัดย่อ

โครงการนี้เป็นการพัฒนาเกมสำหรับการออกกำลังกาย เพื่อเป็นการกระตุ้นให้ผู้เล่นได้ใช้ส่วนต่าง ๆ ของร่างกายในการเล่นเกมนั้น เพื่อให้ผู้เล่นเข้าใจถึงความสำคัญในการออกกำลังกาย การพัฒนาเกมนั้นแบ่งเป็น 2 ส่วนที่ใหญ่ ๆ คือส่วนของฮาร์ดแวร์ และซอฟต์แวร์ ในส่วนสำคัญของฮาร์ดแวร์นั้นคืออุปกรณ์อินพุท สามารถที่ทำให้ผู้เล่นนั้นได้มีการขยับร่างกายในขณะที่เล่นเกม ทางผู้จัดทำจึงได้เลือก แผ่นเกมสำหรับเดินและจอยควบคุมของเครื่องเล่นเกมนิวมาเป็นอินพุทของเกม และในส่วนซอฟต์แวร์นั้นได้ใช้ยูนิตีเกมเอนจินในการออกแบบเกม และใช้เบลนเดอร์ในการออกแบบตัวละคร 3 มิติ เกมนี้จะเป็นเกมแนวไขปริศนาเพื่อผ่านไปด้านต่อไป โดยในแต่ละปริศนานั้นจะออกแบบมาเพื่อให้ผู้เล่นได้ขยับร่างกายให้เหมาะสม และจะสอดแทรกเนื้อหาต่าง ๆ ไว้ภายในเกม รวมถึงแทรกคุณธรรมผ่านเกมด้วย เกมจะใช้มุมมองเป็นแพลตฟอร์ม 3 มิติ พัฒนาลงในระบบปฏิบัติการวินโดวส์

# DEVELOPMENT OF GAME FOR THE EXERCISE

Mr. Napat Ploysomboon 57010656

Mr. Sippakorn Suppapinyo 57011352

Assoc. Prof. Dr. Kietikul Jearanaitannakij Advisor

## ABSTRACT

This project is game that made for exercise. In order to encourage players to use parts of the body to play the game. The development is divided into 2 major parts: hardware and software. The key part of the hardware is input device that can make players to move the body while playing the game. The organizers have chosen the dance pad and wii remote to the game's input device. In part of software used the Unity Game Engine for game design and Blender for design the 3D characters. This game is a puzzle game. Players need to solve the puzzle to go to the next checkpoint. Each puzzle is designed to make players move their body properly. Game will insert content including the moral keep within the game. Game using 3D viewer platform developed in Windows.

## กิตติกรรมประกาศ

สำหรับโครงการพัฒนาเกมเพื่อการออกกำลังกายนี้ที่สามารถสำเร็จลุล่วงได้นั้น ต้องขอขอบคุณ รศ.ดร.เกียรติกุล เจียรนัยชนะกิจ ซึ่งเป็นอาจารย์ที่ปรึกษาของโครงการนี้ ที่ยินดีให้คำปรึกษาและแนะนำในเรื่องต่าง ๆ แม้ในส่วนของโครงการนั้นจะไม่ใช่หัวข้อที่ท่านถนัดก็ตามที่

ขอบพระคุณอาจารย์ทุกท่านที่ได้วิจารณ์ในส่วนต่าง ๆ ของโครงการเพื่อทำให้ผู้จัดทำสามารถพัฒนาโครงการนี้ให้ดียิ่งขึ้น

ขอขอบคุณเพื่อน ๆ และน้อง ๆ ทุกคนที่ช่วยกันให้ความเห็นต่าง ๆ กับเกมเพื่อให้เกมนั้นสามารถตอบโจทย์ในสิ่งที่ต้องการได้ ไม่ว่าจะเป็นการออกกำลังกายและความสนุกสนาน



นภัทร พลอยสมบูรณ์  
สิปปกร ศุภภิญโญ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ .....	III
สารบัญ .....	IV
สารบัญตาราง .....	VI
สารบัญรูป .....	VII
บทที่ 1 บทนำ .....	1
1.1 ความสำคัญและที่มาของโครงการ .....	1
1.2 วัตถุประสงค์ของโครงการ .....	1
1.3 ขอบเขตของโครงการของโครงการ .....	2
1.4 แผนการดำเนินงาน .....	2
1.5 แผนผังการทำงานของโปรแกรมเบื้องต้น .....	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ .....	5
1.7 ส่วนประกอบของปริญญานิพนธ์ .....	6
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง .....	7
2.1 ซอฟต์แวร์ที่ใช้ในการสร้างเกม .....	7
2.2 Script สำหรับควบคุมเกม .....	11
2.3 อุปกรณ์ควบคุมเกม .....	12
2.4 การคำนวณเคลอริ .....	14
บทที่ 3 การออกแบบและพัฒนา .....	15
3.1 ภาพรวมของเกม .....	15
3.2 รูปแบบการเล่น .....	15
3.3 การออกแบบตัวละครและวัตถุภายในเกม .....	16

## สารบัญ (ต่อ)

3.4 การควบคุมภายในเกม .....	21
3.5 การออกกำลังกายภายในเกม .....	24
บทที่ 4 การทดลองและผลการทดลอง .....	26
4.1 การทดลองใช้ซอฟต์แวร์พัฒนาเกม .....	26
4.2 การเขียนโปรแกรมด้วยภาษาซีชาร์ป .....	27
4.3 การออกกำลังกายระหว่างเกม .....	30
บทที่ 5 บทสรุปและข้อเสนอแนะ .....	32
5.1 บทสรุป .....	32
5.2 ปัญหาและอุปสรรค .....	32
5.3 แนวทางการพัฒนาต่อ .....	33



# สารบัญตาราง

ตาราง	หน้า
1.1 แผนการดำเนินการในภาคเรียนที่ 1 .....	3
1.2 แผนการดำเนินการในภาคเรียนที่ 2 .....	3
3.1 การควบคุมพื้นฐานภายในเกม .....	16
4.1 ผลการทดลองการออกกำลังกายระหว่างเล่นเกม 30 นาที .....	30



# สารบัญรูป

รูป	หน้า
1.1 แผนผังการเชื่อมต่ออุปกรณ์ .....	4
1.2 แผนผังแสดงการเชื่อมต่อของซอฟต์แวร์ .....	4
1.3 แผนผังการทำงานเกม .....	5
2.1 สัญลักษณ์ของโปรแกรมยูนิตี้ .....	7
2.2 การใช้งานภายในโปรแกรมยูนิตี้ .....	9
2.3 สัญลักษณ์ของโปรแกรมเบลนเดอร์ .....	9
2.4 การใช้งานภายในโปรแกรมเบลนเดอร์ .....	10
2.5 การใช้งาน Script ในโปรแกรมยูนิตี้ .....	11
2.6 แดนซ์แพด .....	12
2.7 วีรีโมท .....	13
2.8 การเชื่อมต่อวีรีโมท .....	13
3.1 ปริศนาภายในเกม .....	15
3.2 ตัวละครภายในโปรแกรมยูนิตี้ .....	16
3.3 ตัวละครภายในโปรแกรมเบลนเดอร์ .....	17
3.4 วัตถุภายในโปรแกรมยูนิตี้ .....	18
3.5 กล้องภายในโปรแกรมยูนิตี้ .....	18
3.6 ภูมิศาสตร์ภายในโปรแกรมยูนิตี้ .....	19
3.7 ศัตรูภายในโปรแกรมยูนิตี้ .....	20
3.8 อินพุทของแดนซ์แพด .....	22
3.9 การปรับแต่งอินพุทภายในโปรแกรมยูนิตี้ .....	22
3.10 อินพุทของวีรีโมท .....	23
3.11 โปรแกรมสำหรับเชื่อมต่อคอมพิวเตอร์และวีรีโมท .....	23
3.12 ปริศนาภายในเกม .....	25
3.13 แคลอริภายในเกม .....	25
4.1 สร้างวัตถุและภูมิศาสตร์ภายในเกมด้วยโปรแกรมเบลนเดอร์ .....	26
4.2 การนำวัตถุและภูมิศาสตร์สร้างเป็นแผนที่ในโปรแกรมยูนิตี้ .....	27

## สารบัญรูป (ต่อ)

4.3 การเดินภายในเกม .....	28
4.4 การปรับทิศทางตัวละครภายในเกม .....	28
4.5 การจับภาพภาพของกล้อง .....	29
4.6 ด้านผลักกล้องไปทับสวิทช์เพื่อเป็ประตุ .....	29
4.7 ศัตรูที่เกิดขึ้นมาก่อกวนผู้เล่นภายในเกม .....	30



# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของโครงการ

ในยุคปัจจุบันนี้มนุษย์ได้มีสื่อต่าง ๆ ที่เข้ามาในชีวิตประจำวัน และเกมถือเป็นสิ่งที่มีผลต่อชีวิตประจำวันของมนุษย์มาก เนื่องด้วยในตอนนี้เทคโนโลยีที่เข้ามานั้นทำให้ความสะดวกสบายต่าง ๆ เพิ่มขึ้นจากที่เมื่อก่อนเกมยังเข้าถึงยาก เนื่องจากอุปกรณ์ที่ต้องใช้ในการเล่นเกมไม่ได้แพร่หลายเหมือนทุกวันนี้ ในสมัยนี้อุปกรณ์ที่ใช้ในการเล่นเกมนั้นมีมากมาย ทำให้มนุษย์สามารถเข้าถึงเกมได้ในทุกช่วงอายุโดยส่วนใหญ่จะเป็นเด็กและเยาวชน จนปัจจุบันแทบจะสามารถกล่าวได้ว่าเกมเป็นส่วนหนึ่งของชีวิต ทุก ๆ วันนี่เกมสามารถเข้าถึงได้ไม่ยากเหมือนแต่ก่อนแล้ว จึงถือเป็นเรื่องสำคัญในการที่เราจะต้องเลือกสรรเกมที่ดีและมีประโยชน์ต่อผู้เล่น ทางผู้จัดทำโครงการ ได้เล็งเห็นถึงความสำคัญในจุดนี้ จึงคิดที่จะพัฒนาเกมที่สร้างสรรค์และเป็นประโยชน์ต่อเยาวชน

เกมในปัจจุบันมีมากมายหลากหลายรูปแบบ แต่สิ่งหนึ่งที่เกมส่วนใหญ่มีเหมือนกัน ก็คือการนั่งเล่นอยู่กับที่ ทำให้สุขภาพร่างกายของผู้เล่นนั้นเสื่อมถอยไปอย่างช้า ๆ ทางผู้จัดทำโครงการจึงต้องการที่จะพัฒนาเกม ที่ผู้เล่นต้องมีการขยับร่างกายไปด้วย เพื่อใช้ในการควบคุมระบบการทำกิจกรรมต่าง ๆ ภายในเกม โดยมีเป้าหมายที่จะส่งเสริมการออกกำลังกายให้แก่ผู้เล่น

### 1.2 วัตถุประสงค์โครงการ

- 1) เพื่อเป็นการศึกษาและพัฒนาเกมโดยใช้ Unity - Game Engine
- 2) เพื่อเป็นการศึกษาการสร้างภาพคอมพิวเตอร์กราฟฟิกและภาพเคลื่อนไหวเพื่อนำมาใช้ในเกม
- 3) เพื่อเป็นการพัฒนาเกมที่ผู้เล่นสามารถทำการบริหารร่างกายไปด้วยได้
- 4) เพื่อเป็นการศึกษาและประยุกต์ใช้เซนเซอร์เป็นอุปกรณ์รับค่าอินพุตการเคลื่อนไหวของผู้เล่น ที่นำมาใช้ในการควบคุมเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ขอบเขตของโครงการ

- 1) เป็นเกมแนว Puzzle ,Platform 3 มิติ
- 2) เล่นบนระบบปฏิบัติการ Windows
- 3) ใช้ Unity- Game Engine ในการพัฒนาเกม
- 4) ระบบควบคุมเกมทำงานได้ตามที่ออกแบบไว้
- 5) เป็นการพัฒนาเกมที่สามารถมีผู้เล่นได้เพียงคนเดียว
- 6) มีการรับอินพุตผ่านอุปกรณ์ต่าง ๆ
  - 6.1) แผ่นเกมสำหรับเกมเต้น (Dance Pad)
  - 6.2) จอยควบคุมของเครื่องเล่นเกม Wii (Wii Remote)

### 1.4 แผนการดำเนินงาน

- 1) กำหนดขอบเขต เนื้อหาของเกม กติกาในการเล่น
- 2) ศึกษาข้อมูลวิธีการใช้งาน โปรแกรม Unity ซึ่งเป็น โปรแกรมที่ใช้ในการพัฒนาเกมที่ผู้จัดทำได้เลือกมา
- 3) ศึกษาข้อมูลวิธีการใช้งานโปรแกรมสำหรับออกแบบตัวละคร 3 มิติ Blender
- 4) ศึกษาข้อมูลของอุปกรณ์ที่ใช้ในการรับส่งข้อมูลผ่านเซนเซอร์ เพื่อเลือกใช้อุปกรณ์สำหรับโครงการ
- 5) ออกแบบเกมและส่วนประกอบทั้งหมดของเกม
- 6) พัฒนาเกมตามที่ได้ออกแบบไว้
- 7) ทดสอบและแก้ไขข้อบกพร่องต่าง ๆ ที่เกิดขึ้นระหว่างที่เล่นเกม แล้วปรับปรุงเกมให้ดีขึ้นและเพื่อให้ เกมมีการทำงานที่สมบูรณ์ที่สุด
- 8) ทดสอบการเชื่อมต่อการส่งและรับค่าระหว่างอุปกรณ์และเกม
- 9) สรุปผลโครงการ
- 10) จัดทำเอกสารและรวบรวมความรู้ทั้งหมด

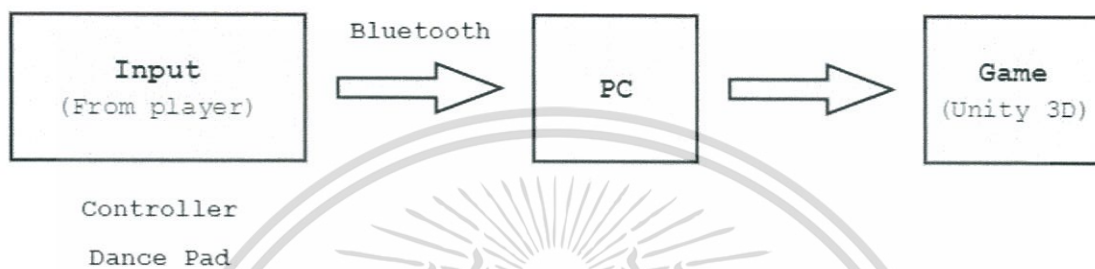
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## 1.5 แผนผังการทำงานของโปรแกรมเบื้องต้น

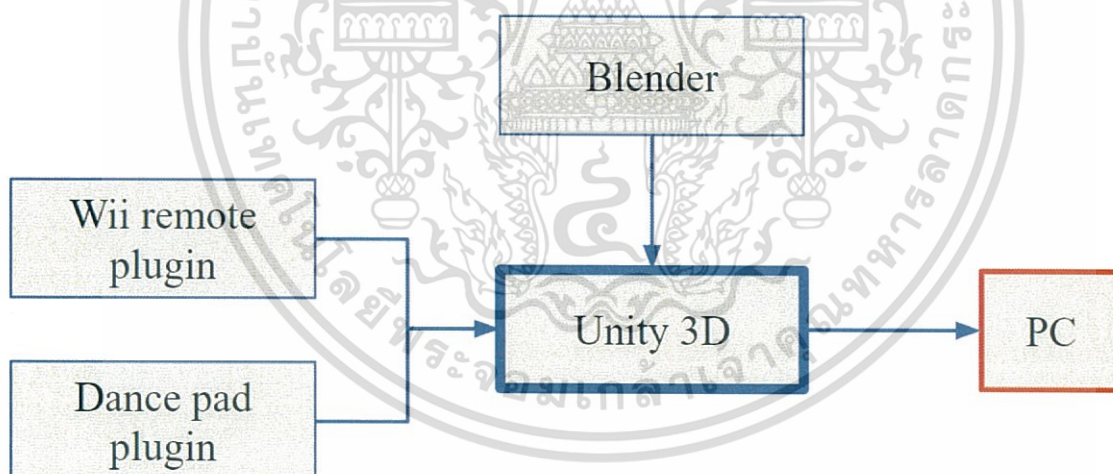
### 1.5.1 แผนผังแสดงการเชื่อมต่อของอุปกรณ์

- 1) Input ที่ได้จากผู้เล่นได้แก่ Controller, Dance Pad
- 2) PC คอมพิวเตอร์ ในที่นี้จะใช้ระบบปฏิบัติการ Windows
- 3) Game ที่สร้างโดยใช้ Unity (3D) - Game Engine



รูปที่ 1.1 แผนผังการเชื่อมต่ออุปกรณ์

### 1.5.2 แผนผังแสดงการเชื่อมต่อของซอฟต์แวร์

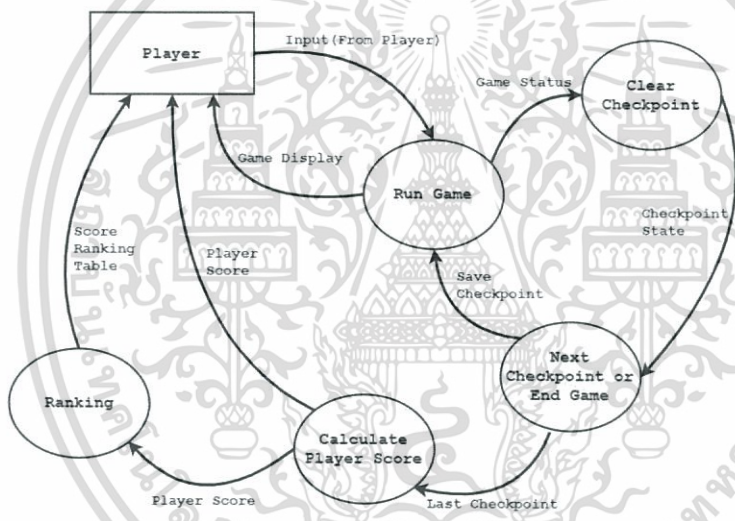


รูปที่ 1.2 แผนผังแสดงการเชื่อมต่อของซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.5.3 แผนผังแสดงการทำงานของเกม

- 1) Run Game จะรับค่าอินพุตจากผู้เล่น โดยแสดงค่าเอาท์พุทผ่านทางหน้าจอ
- 2) Clear Checkpoint เป็นส่วนที่เช็คว่าผู้เล่นผ่านสเตจไหนไปแล้วบ้าง จากนั้นก็จะไปสเตจถัดไปหากผู้เล่นได้ผ่านสเตจแล้ว
- 3) Next Checkpoint or End Game ถ้าผู้เล่นยังไม่ถึงสเตจสุดท้ายก็ให้ไปสเตจถัดไปจากนั้นก็ Save ไว้หากผู้เล่นไม่ผ่านก็จะสามารถย้อนมายังจุด Checkpoint ในสเตจเดิมได้
- 4) Calculate Player Score เป็นส่วนที่คำนวณคะแนนของผู้เล่นเมื่อเล่นถึงสเตจสุดท้ายแล้ว โดยจะแสดงคะแนนไปยังผู้เล่น
- 5) Ranking ใช้ในการจัดอันดับของผู้เล่นที่ทำคะแนนได้ดีที่สุด โดยผู้เล่นจะสามารถดูผลลัพธ์ที่ได้ในรูปแบบของตาราง



รูปที่ 1.3 แผนผังการทำงานของเกม

### 1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้พัฒนาทักษะต่างๆและกระบวนการที่ได้ใช้ระหว่างการพัฒนาเกม
- 2) ได้พัฒนาทักษะการทำกราฟฟิคดีไซน์จากการออกแบบเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.7 ส่วนประกอบของปริญญานิพนธ์

บทที่ 1 บทนำ กล่าวถึงที่มาและความสำคัญของโครงการ วัตถุประสงค์ของโครงการ ขอบเขตของโครงการ แผนผังการทำงานของโปรแกรมเบื้องต้น ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับโครงการ กล่าวถึงซอฟต์แวร์ที่ใช้ในการเกมขึ้นมา ยูนิคซ์และเบลนเดอร์ และกล่าวถึงฮาร์ดแวร์ที่ใช้เป็นอินพุทของเกม แดนซ์แพคและจอยควบคุมของเครื่องเกมวี

บทที่ 3 การออกแบบและพัฒนาโครงการ กล่าวถึงการออกแบบเกมโดยภาพรวม

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการทดลองเล่นเกมเพื่อดูผลว่าจากการเล่นเกม มีผลต่อการออกกำลังกายของผู้เล่นในด้านใดบ้าง ผู้เล่นได้ใช้พลังงานไปเท่าใดในการเล่น

บทที่ 5 บทสรุป กล่าวถึงข้อสรุปต่าง ๆ ภายในโครงการ ข้อจำกัด ปัญหาและอุปสรรคต่าง ๆ ของโครงการ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อยอด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้อง

### 2.1 ซอฟต์แวร์ที่ใช้ในการสร้างเกม

#### 2.1.1 Unity

ยูนิตี้ (Unity) หรือที่เรียกว่า Unity Technologies มีแพลตฟอร์มสำหรับการสร้างเกมและแอปพลิเคชันที่รองรับการแสดงผลในรูปแบบ 2D (2-dimensions), 3D (3-dimensions), VR (Virtual Reality) และ AR (Augmented Reality Technology) และเป็นเครื่องมือกราฟิกที่มีประสิทธิภาพสูง รวมทั้งเป็นโปรแกรม Editor ที่สามารถช่วยให้ผู้ใช้งานสามารถสร้างผลงานอันสร้างสรรค์และนำเสนอออกมาได้อย่างรวดเร็วบนสื่อหรืออุปกรณ์ต่างๆ ผู้ใช้งาน Unity สามารถเชื่อมต่อกับผู้ชมบนคอมพิวเตอร์ส่วนบุคคล (PC), คอนโซล (Console), เว็บ (Web), โทรศัพท์มือถือ (Mobile Device), ระบบความบันเทิงภายในบ้าน (Home Entertainment Systems), ระบบฝังตัว (Embedded Systems) หรือจอแสดงผลที่ติดตั้งบนหัว (Head-Mounted Displays) ได้อย่างง่ายดาย



รูป 2.1 สัญลักษณ์ของโปรแกรมยูนิตี้

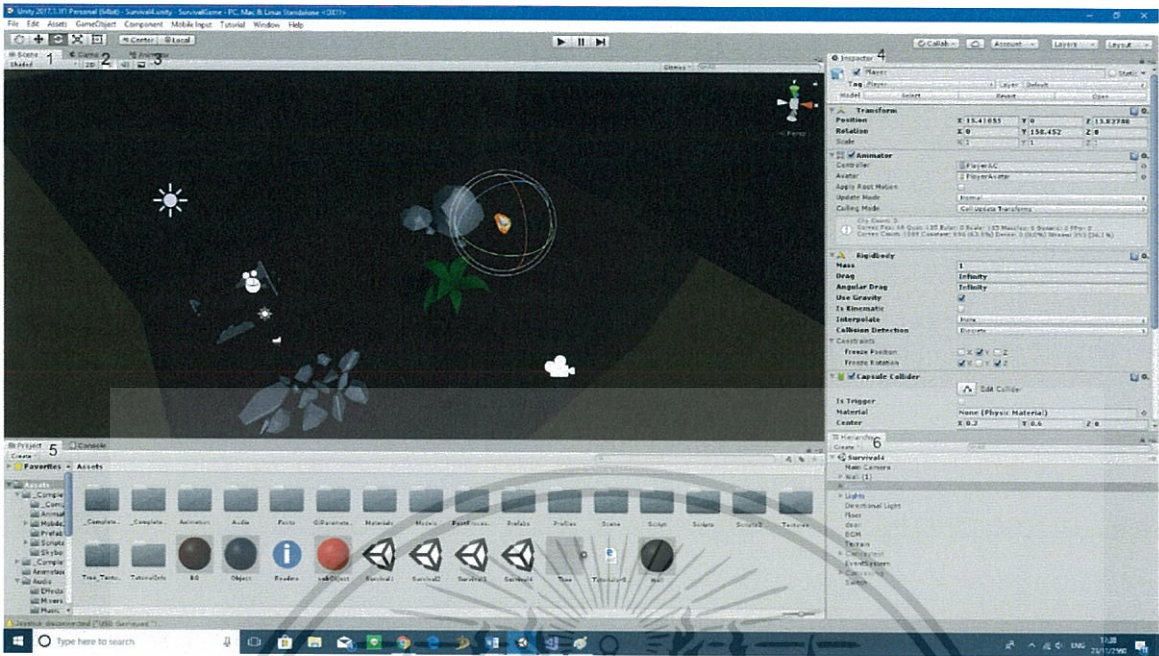
ยูนิตี้เป็นมากกว่าเกมเอนจิน สามารถช่วยผู้ใช้งานให้ประสบความสำเร็จได้อย่างต่อเนื่อง มีทุกอย่างที่ต้องการเพื่อพัฒนาผลงานที่มีคุณภาพซึ่งสามารถเพิ่มประสิทธิภาพรวมทั้งเชื่อมต่อกับผู้ชมด้วยเครื่องมือ (Tools) และทรัพยากร (resources) ที่ประกอบด้วย Unity Asset Store, Unity Cloud Build, Unity Analytics, Unity Ads, Unity Every-play และ Unity Certification

Unity Technologies มีผู้ใช้งานมากกว่าหนึ่งล้านคนที่เป็นทั้งนักพัฒนาซอฟต์แวร์, สตูดิโอขนาดใหญ่, สตูดิโอขนาดเล็ก (Indie Studios), นักศึกษา หรือแม้กระทั่งมือสมัครเล่นทั่วโลก จึงทำให้สามารถเข้าถึงยูนิตี้ได้ทุกเพศทุกวัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าต่างการใช้งานของโปรแกรมยูนิตีนั้น จะประกอบไปด้วยส่วนหลัก ๆ ดังนี้

- 1) Scene เป็นส่วนหน้าจอสำหรับพัฒนาเกม ผู้พัฒนาเกมสามารถนำวัตถุที่สร้างไว้มาใส่ไว้ในส่วนนี้ เพื่อใช้เป็นวัตถุภายในเกม
- 2) Game เป็นส่วนที่แสดงหน้าจอขณะเล่นเกม โดยหน้านี้จะแสดงผลที่ต่อเมื่อผู้พัฒนาเกมใช้ตัว Main Camera ภายในเกม ภาพที่ออกมาจะเป็นภาพที่เห็นในมุมมองของ Main Camera
- 3) Animator เป็นส่วนสำหรับสร้างระบบอนิเมชัน ผู้พัฒนาเกมจะสามารถนำอนิเมชันต่าง ๆ ที่สร้างไว้ มาทำการเชื่อมต่อกันเพื่อทำให้วัตถุหรือตัวละครนั้น มีกิริยาท่าทาง ๆ ตามที่ได้ออกแบบไว้
- 4) Inspector เป็นหน้าต่างแสดงส่วนประกอบ (Component) ต่าง ๆ ของวัตถุ (Object) เป็นส่วนที่ผู้สร้างเกมสามารถใส่ส่วนประกอบต่าง ๆ เข้าไปในวัตถุที่ตนเองเลือก โดยส่วนประกอบภายในโปรแกรมยูนิตีนั้นค่อนข้างมาก แต่ส่วนประกอบที่สำคัญในการพัฒนาเกมในโครงการนี้นั้นคือ
  - 4.1) Transform คือส่วนประกอบที่ใช้ในการกำหนดตำแหน่ง องศา และขนาดของวัตถุ
  - 4.2) Animator คือส่วนประกอบที่ใช้ในการทำให้วัตถุนั้นสามารถเคลื่อนไหวเป็นอนิเมชันได้
  - 4.3) Rigidbody คือส่วนประกอบที่ทำให้วัตถุนั้น ๆ มีระบบฟิสิกส์เป็นของตัวเอง
  - 4.4) Collider คือส่วนประกอบที่ใช้ในการกำหนดว่าวัตถุที่ทับซ้อนทับ Collider นั้นคือส่วนที่สามารถจับต้องได้ Collider จะมีหลายรูปแบบ ผู้พัฒนาเกมจะต้องใช้รูปแบบให้เหมาะสมกับวัตถุของตนเอง
  - 4.5) Script คือส่วนประกอบที่สร้างขึ้นจากภาษา C# มีส่วนใช้ในการควบคุมวัตถุต่าง ๆ ให้เป็นไปตามที่ผู้พัฒนาเกมได้เขียน โปรแกรมไว้
- 5) Project เป็นหน้าต่างแสดงไฟล์ต่าง ๆ ที่อยู่ในโฟลเดอร์ Project ที่เราเปิดอยู่ ผู้พัฒนาเกมจะต้องเก็บข้อมูลต่าง ๆ ไว้ในโปรเจกต์ของตนเองผ่านโฟลเดอร์ที่ชื่อว่า Assets
- 6) Hierarchy เป็นหน้าต่างแสดงว่ามีวัตถุอะไรบ้างในเราใส่ไว้ใน Scene ที่เราเปิดอยู่ เมื่อผู้พัฒนาเกมได้นำวัตถุต่าง ๆ เข้ามาใส่ไว้ใน Scene วัตถุนั้นจะถูกเพิ่มขึ้นมาใน Hierarchy เพื่อให้ผู้พัฒนาเกมสะดวกในการเลือกวัตถุนั้น



รูป 2.2 การใช้งานภายในโปรแกรมยูนิตี

### 2.1.2 Blender

Blender เป็นโปรแกรมฟรีและ โปรแกรมสร้างวัตถุ 3 มิติแบบ โอเพนซอร์ส (Open Source) Blender รองรับการสร้างกราฟฟิกแบบ ไปป์ไลน์โมเดลลิ่ง (Pipeline-Modeling), เสื้อผ้า (Rigging), อนิเมชัน (Animation) การสร้างภาพจำลอง (Simulation), การสร้างภาพจากแบบจำลอง (Rendering), คอมโพสิต (Compositing) และการติดตามการเคลื่อนไหว (Motion Tracking) แม้แต่การตัดต่อวิดีโอและการสร้างเกมก็สามารถทำได้ ผู้เชี่ยวชาญใช้งาน API ของ Blender ในการสร้างสคริปต์ Python เพื่อปรับแต่งแอปพลิเคชันและเขียนเครื่องมือพิเศษขึ้นมา โดยเครื่องมือที่ได้เขียนขึ้นมานั้นมักจะถูกนำมาใส่ไว้ในเวอร์ชันในอนาคต Blender เหมาะกับบุคคลและสตูดิโอขนาดเล็กที่ได้รับประโยชน์จากไปป์ไลน์และกระบวนการพัฒนาที่เป็นแบบ Responsive



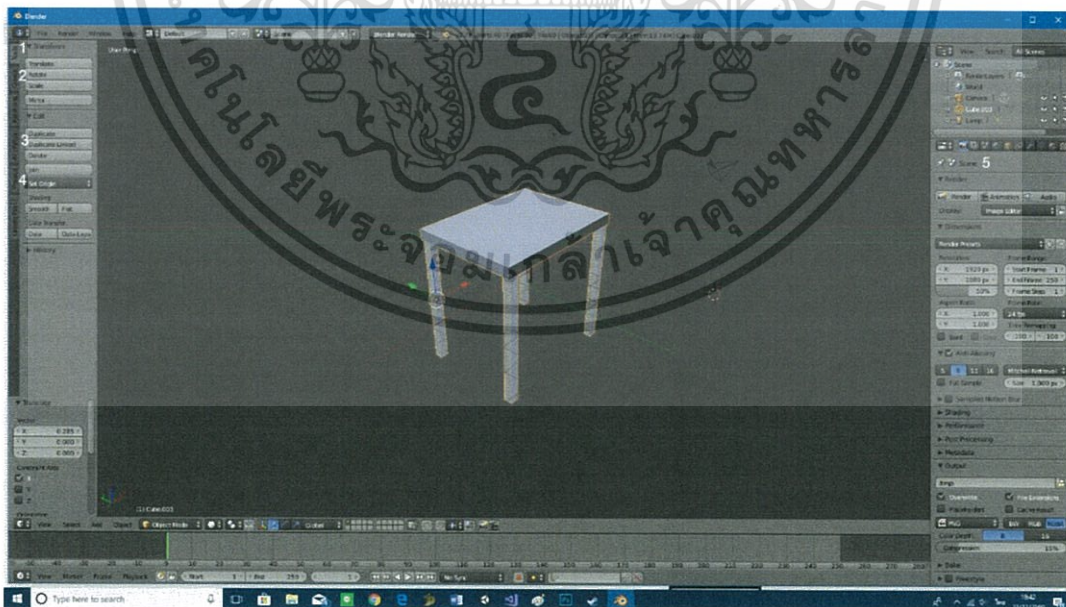
รูป 2.3 สัญลักษณ์ของโปรแกรมเบเลนเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Blender เป็นโปรแกรมแบบข้ามแพลตฟอร์ม (Cross-Platform) และทำงานได้ดีพอ ๆ กันบนเครื่อง Linux, Windows และ Macintosh Blender มีใบอนุญาต General Public License (GPL) ซึ่งทำให้บุคคลทั่วไปสามารถเปลี่ยนแปลงฐานข้อมูลโค้ดขนาดเล็กและใหญ่ ซึ่งนำไปสู่การแก้ไขข้อบกพร่องและการใช้งานที่ดียิ่งขึ้น โดยการตั้งชุมชนในการแลกเปลี่ยนความคิดเห็นมา (Community) Blender ทำให้ทุกคนเสมือนมีซอฟต์แวร์ 3 มิติเป็นของตัวเอง

หน้าต่างการใช้งานของโปรแกรมเบลนเดอร์นั้น จะประกอบไปด้วยส่วนหลัก ๆ ดังนี้

- 1) Tools เป็นส่วนคำสั่งที่ใช้สำหรับการจัดการ ตำแหน่ง มุม และขนาด รวมถึงคำสั่งในการทำซ้ำของวัตถุที่เราสนใจ
- 2) Create เป็นส่วนคำสั่งสำหรับใช้สร้างวัตถุต่าง ๆ มีที่ให้อยู่แล้วภายในโปรแกรมเบลนเดอร์ ซึ่งส่วนใหญ่มักจะเป็นวัตถุพื้นฐาน
- 3) Animation เป็นส่วนสำหรับการสร้างอนิเมชันภายในโปรแกรมเบลนเดอร์
- 4) Physics เป็นส่วนสำหรับใช้จัดการระบบฟิสิกส์ของวัตถุ
- 5) Scene คือส่วนแสดงผลการสร้างโดยภาพรวมของโปรแกรมเบลนเดอร์ ที่มีไว้ใช้ดูว่าเมื่อเราสร้างวัตถุออกมาแล้ว รูปร่างของวัตถุเป็นอย่างไร สามารถใช้กล้องเพื่อดูวัตถุว่าเมื่อมองจากมุมมองต่าง ๆ แล้ววัตถุจะเป็นเช่นไร และสามารถปรับขนาดของกล้องที่ต้องการได้ในส่วนนี้



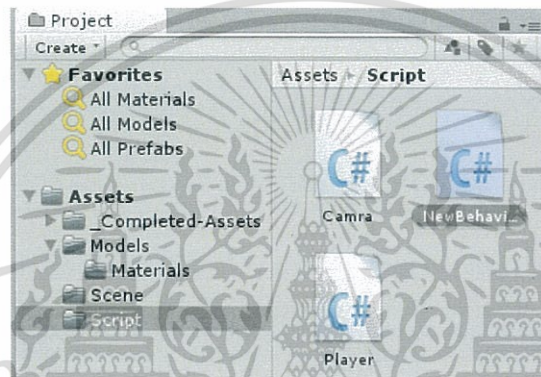
รูป 2.4 การใช้งานภายในโปรแกรมเบลนเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 Script

เป็นส่วน Programming ที่ใช้กับตัวเกมโดยพื้นฐานแล้วใช้ C# เป็นหลักในการเขียนและจะมีการใช้ Unity-Script ซึ่งเป็นภาษาที่ออกแบบมาโดยเฉพาะสำหรับการใช้งานใน Unity

การทำงานของ Game-Objects ถูกควบคุมโดยคอมโพเนนต์ (Components) ซึ่งอยู่ในสคริปต์ (Script) เหล่านี้ แม้ว่าคอมโพเนนต์ภายในของ Unity สามารถใช้งานได้หลากหลาย แต่ในการทำระบบเล่นเกมยังจำเป็นต้องใช้สคริปต์เพื่อการทำงานที่หลากหลายมากขึ้น Unity ช่วยให้คุณสามารถสร้างคอมโพเนนต์ของคุณเองโดยการใช้สคริปต์ แก้ไขคุณสมบัติของคอมโพเนนต์ตามเวลาและการตอบสนองต่ออินพุทของผู้ใช้ในแบบที่ต้องการ



รูป 2.5 การใช้งาน Script ในโปรแกรมยูนิตี

### โปรแกรม 1.1 ตัวอย่างการเขียน Script ของเกม

```
using UnityEngine;
using System.Collections;

public class MainPlayer : MonoBehaviour {
    // Use this for initialization
    void Start () {
    }

    // Update is called once per frame
    void Update() {
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 อุปกรณ์สำหรับควบคุมเกม (Game Controller)

ในส่วนของ Game Controller ที่ใช้ในการทำโครงงานนั้น ทางผู้จัดทำได้นำมาใช้อยู่ 2 ชนิดได้แก่ Dance Pad และ Wii Remote

### 2.3.1 Dance Pad



รูป 2.6 แดนซ์แพด

Dance Pad คือ อุปกรณ์อิเล็กทรอนิกส์แบบแบนซึ่งใช้เป็นอินพุตในเกมประเภทเต้น โดยใช้ในการเชื่อมต่ออุปกรณ์กับคอมพิวเตอร์แบบ USB ซึ่งทางผู้จัดทำได้นำเอาแผ่นเกมเต้น (Dance Pad) มาประยุกต์ใช้กับทางโครงงาน เพื่อที่จะสนับสนุนให้ผู้เล่นได้มีการเคลื่อนไหวร่างกายทำให้การเล่นเกมนี้ก็สามารถออกกำลังกายได้

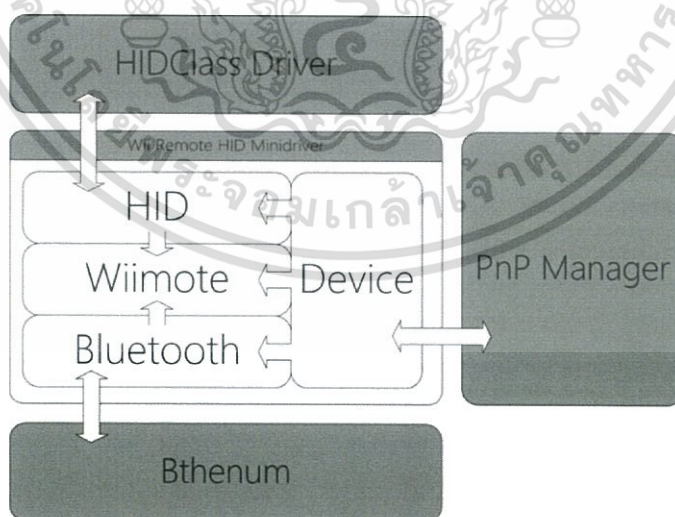
### 2.3.2 Wii Remote

Wii remote หรือ Wii-mote เป็นที่คอนโทรลเลอร์ของเครื่องวีทีเอไอวีที่ใช้ในเกมหลัก ๆ ของเครื่องวี Wii-mote มีรูปร่างเหมือนรีโมตโทรทัศน์ โดยมีคุณสมบัติหลักคือมีเซนเซอร์ในการตรวจจับการเคลื่อนไหว ซึ่งใช้ในการควบคุมเกมโดยการเคลื่อนไหวรีโมตนี้ไปในทิศทางต่าง ๆ และต้องวางตัวตรวจจับความเคลื่อนไหว (Wii Sensor Bar) ไว้ด้านบนของโทรทัศน์ที่ใช้เล่น ซึ่งโดยปกติแล้วจะใช้ Wii-mote กับเครื่องวีเท่านั้น แต่ทางผู้จัดทำได้นำ Wii-mote มาเชื่อมต่อกับคอมพิวเตอร์เพื่อที่จะใช้ในการเล่นเกมบนคอมพิวเตอร์ ซึ่งจะใช้กับระบบปฏิบัติการ Windows



รูป 2.7 วีรีโมท

โดยปกติแล้วการเชื่อมต่อ Wii remote กับเครื่อง PC ที่ใช้ระบบปฏิบัติการ Windows นั้นมีปัญหาที่คือ Windows ไม่สามารถรู้ว่าเป็นพอร์ทจาก Wii remote นั้นคืออะไร จึงต้องมีการลง Driver เพื่อเชื่อมต่ออุปกรณ์เข้าด้วยกัน โดยจะใช้ Driver ที่มีชื่อว่า HID WIIMOTE โดยวิธีแก้ปัญหาก็คือนำ Driver ไปใส่ระหว่าง Wii remote กับ HID class driver เพื่อที่จะแก้ไขในส่วนของการเชื่อมต่อกันกับอุปกรณ์ ไดรเวอร์ HID Class ถูกจับคู่กับ Mini driver ซึ่ง Mini driver ทำหน้าที่จัดการการเชื่อมต่อที่เฉพาะเจาะจงในการส่งข้อมูลกับอุปกรณ์



รูป 2.8 การเชื่อมต่อวีรีโมท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Plug and Play Manager (PnP Manager) สนับสนุนฟังก์ชันการทำงานของ PnP ใน Windows และรับผิดชอบงานที่เกี่ยวข้องกับ PnP ต่อไปนี้

- 1) การตรวจนับและนับจำนวนอุปกรณ์ในขณะที่ระบบกำลังบูท
- 2) เพิ่มหรือลบอุปกรณ์ในขณะที่ระบบกำลังทำงานอยู่

## 2.4 การคำนวณแคลอรี (Calories Calculation)

ในการคำนวณแคลอรีภายในเกมนั้นแบ่งออกเป็น 2 ประเภทได้แก่ การคำนวณแคลอรีของขา และการคำนวณแคลอรีของแขน ในการตรวจสอบหากเราขยับขานั้นจะได้เผาผลาญพลังงานกี่แคลอรีนั้นทางผู้จัดทำโครงการได้ทำการทดลองเดินเป็นระยะทาง 10,000 ก้าว โดยใช้แอปพลิเคชันสำหรับออกกำลังกายต่าง ๆ ตรวจสอบดูแล้วนำมาประมาณค่า โดยสรุปแล้วเราจะใช้การเดิน 10,000 ก้าวนั้นจะเผาผลาญพลังงานประมาณ 500 แคลอรี โดยการคิดว่าการเดินนั้น ความจริงเราได้ใช้ทุกส่วนของร่างกาย แต่เนื่องจากการเดินนั้น สิ่งที่ยับมากที่สุดเป็นส่วนขา เราจึงได้ประมาณแบ่งการให้การขยับขานั้นหากขยับ 10,000 ก้าวจะเผาผลาญที่ 400 แคลอรี ซึ่งการขยับ 1 ก้าวนั้นก็เผาผลาญแคลอรีไป 0.04 แคลอรี และในส่วนของแขนนั้นผู้จัดทำโครงการได้ใช้การจับเวลาที่เราได้เล่นเกมเป็นหน่วยวินาที และคิดเป็นวินาทีละ 0.007 แคลอรี และเพิ่มการคำนวณจากการนำจำนวนครั้งของการขยับขา มาเพิ่มให้แคลอรีที่ทำการเผาผลาญในส่วนของแขนด้วย โดยใช้ตัวคูณเป็น 0.003 โดยได้สมการเป็น

$$\text{แคลอรีที่เผาผลาญจากแขน} = 0.007 * \text{วินาทีที่ทำการเล่นเกม} + 0.003 * \text{จำนวนการขยับขา}$$

## บทที่ 3

### การออกแบบและพัฒนา

#### 3.1 ภาพรวมของเกม

เป็นเกมที่มีผู้เล่นเพียงคนเดียว เป็นเกมแนวไขปริศนา (Puzzle) เพื่อผ่านไปยังด่านต่อไป โดยใช้มุมมองเป็นแบบ Platform 3D Side-Scrolling โดยมีเป้าหมายที่จะผลิตลงใน Platform PC (Windows Operation System) โดยมีเป้าหมายให้ผู้เล่นเกมนั้นได้ออกกำลังกายไปด้วย

#### 3.2 รูปแบบการเล่น

ผู้เล่นจะต้องแก้ไขปริศนาต่าง ๆ ที่มีในแต่ละด่าน โดยจะมีข้อจำกัดภายในด่านต่างกันไป ซึ่งก็จะมีผลให้ความยากของแต่ละด่านแตกต่างกันออกไปด้วย ตัวอย่างเช่น ปริศนาภายในเกมเล่นการผลักกล่องสีเหลี่ยมไปทับสวิตช์เพื่อเปิดประตูดังรูป 3.1



รูป 3.1 ปริศนาภายในเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 การควบคุมภายในเกม

#### ตารางที่ 3.1 การควบคุมพื้นฐานภายในเกม

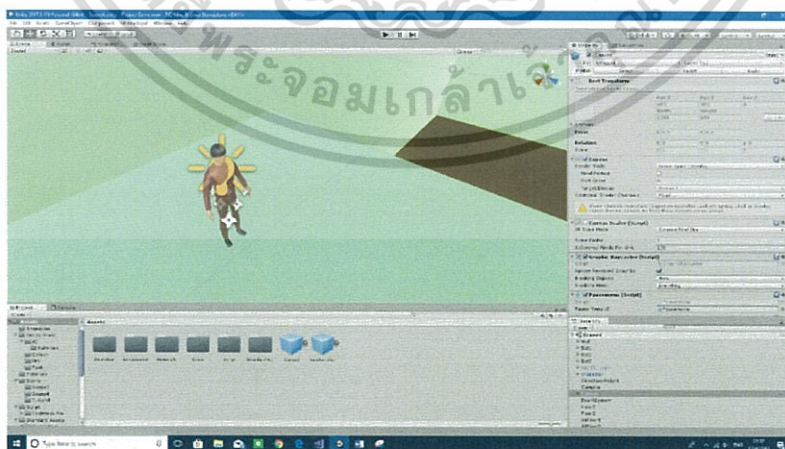
การกระทำ (Action)	การควบคุม (Control)	อินพุท (Input)
เดิน	ผู้เล่นจะต้องเหยียบ Dance pad เพื่อให้ตัวละครเคลื่อนที่	Dance pad
หมุนตัวละคร	หมุนวีรี โมทเพื่อทำการหมุนตัวละคร	Wii remote

### 3.3 การออกแบบตัวละครและวัตถุภายในเกม

ในการสร้างตัวละครหรือวัตถุต่าง ๆ ภายในเกมนั้น ในโปรแกรมยูนิตี้ จะมองมันเป็นวัตถุ (Object) โดยแต่ละวัตถุก็จะสามารถใส่ข้อมูลต่าง ๆ ลงไปได้ เพื่อให้วัตถุนั้นทำสิ่งต่าง ๆ ได้แตกต่างกัน โดยเราเรียกข้อมูลที่ใส่เข้าไปว่าส่วนประกอบ (Component) และสามารถนำวัตถุแต่ละชนิดเชื่อมต่อกับ Script เพื่อควบคุมมันด้วยเงื่อนไขต่าง ๆ ที่เราสร้างใน Script ได้อีกด้วย โดยตัว Script ก็ถือเป็นส่วนประกอบชนิดหนึ่งเช่นกัน

#### 3.3.1 ตัวละคร

การออกแบบตัวละครที่ใช้ในการเล่นเกมนั้นจะออกแบบโดยใช้โปรแกรม Blender โดยจะมีส่วนประกอบที่สำคัญสำหรับตัวละครนั้นซึ่งทำการอิมพอร์ตจากโปรแกรมเบสเนเจอร์เข้าไปในโปรแกรมยูนิตี้มี ดังนี้



รูป 3.2 ตัวละครภายในโปรแกรมยูนิตี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) Transform คือส่วนประกอบที่ใช้ในการกำหนดตำแหน่ง องศา และขนาดของวัตถุ
- 2) Animator คือส่วนประกอบที่ใช้ในการทำให้วัตถุนั้นสามารถทำอนิเมชันได้
- 3) Rigidbody คือส่วนประกอบที่ทำให้วัตถุนั้นมีระบบฟิสิกส์เป็นของตนเอง
- 4) Capsule Collider คือส่วนประกอบที่ใช้ในการกำหนดว่าวัตถุที่ทับซ้อนทับ Collider นั้นคือส่วนที่สามารถจับต้องได้ Collider จะมีหลายรูปแบบ ผู้พัฒนาเกมจะต้องใช้รูปแบบให้เหมาะสมกับวัตถุของตนเอง โดยในที่นี้ใช้เป็นรูปแบบ Capsule
- 5) Script Player Movements เป็น Script สำหรับใช้ในการควบคุมการเคลื่อนที่ของตัวละครและการทำแอคชั่นต่าง ๆ

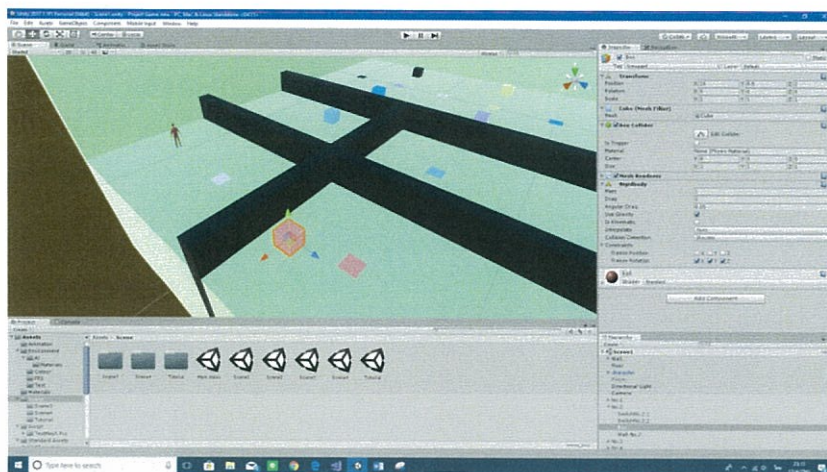


รูป 3.3 ตัวละครภายในโปรแกรมเบลนเดอร์

### 3.3.2 วัตถุภายในเกม

การออกแบบวัตถุภายในเกมนั้น จะมีส่วนประกอบที่สำคัญสำหรับวัตถุ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

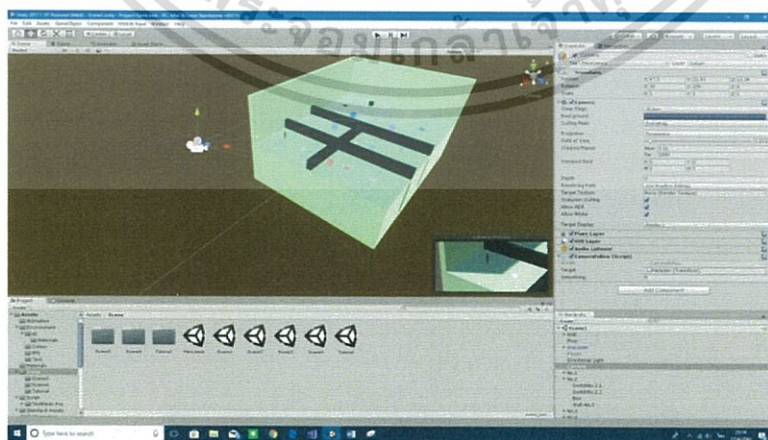


รูป 3.4 วัตถุภายในโปรแกรมยูนิตี

- 1) Transform คือส่วนประกอบที่ใช้ในการกำหนดตำแหน่ง องศา และขนาดของวัตถุ
- 2) Rigidbody คือส่วนประกอบที่ทำให้วัตถุนั้นมีระบบฟิสิกส์เป็นของตนเอง
- 3) Collider คือส่วนประกอบที่ใช้ในการกำหนดว่าวัตถุที่ทับซ้อนทับ Collider นั้นคือส่วนที่สามารถจับต้องได้ Collider จะมีหลายรูปแบบ ผู้พัฒนาเกมจะต้องใช้รูปแบบให้เหมาะสมกับวัตถุของตนเอง
- 4) Script ต่าง ๆ เพื่อใช้ในการควบคุมวัตถุที่จำเป็น

### 3.3.3 กล้องติดตามตัวละคร

การออกแบบกล้องติดตามตัวละครภายในเกมนั้น จะมีส่วนประกอบที่สำคัญสำหรับกล้องติดตามตัวละคร ดังนี้



รูป 3.5 กล้องภายในโปรแกรมยูนิตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

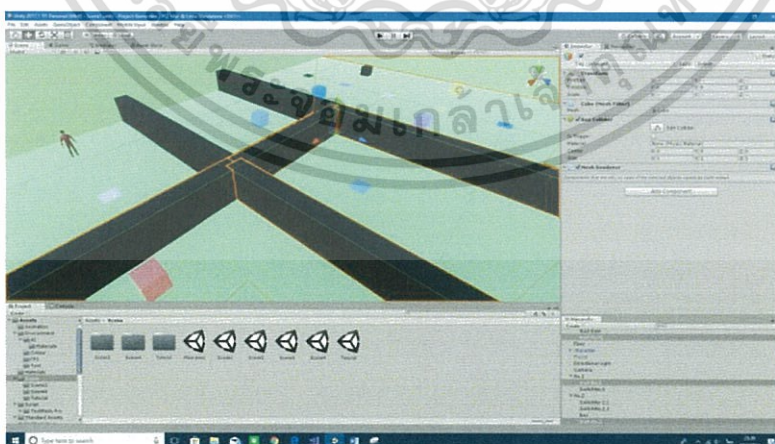
- 1) Transform คือส่วนประกอบที่ใช้ในการกำหนดตำแหน่ง องศา และขนาดของวัตถุ
- 2) Rigidbody คือส่วนประกอบที่ทำให้วัตถุนั้นมีระบบฟิสิกส์เป็นของตนเอง
- 3) Script ต่าง ๆ เพื่อใช้ในการควบคุมตำแหน่งของกล้อง โดยกำหนดให้กล้องจะเปลี่ยนตำแหน่งตามตำแหน่งของตัวละครที่เปลี่ยนไป

### โปรแกรม 3.1 การเขียน Script การเคลื่อนที่ของกล้องจับภาพตัวละคร

```
void Start() {
    offset = transform.position - target.position;
}
Void Update() {
    Vector 3 targetCampos = target.position + offset;
    Transform.position = Vector3.Lerp(transform.position,
    targetsCampos,smoothing*Time.deltaTime);
} // target.position คือตำแหน่งของตัวละคร
```

#### 3.3.4 ภูมิศาสตร์ภายในเกม

การออกแบบภูมิศาสตร์ภายในเกมนั้น จะมีส่วนประกอบที่สำคัญสำหรับภูมิศาสตร์ภายในเกม ดังนี้



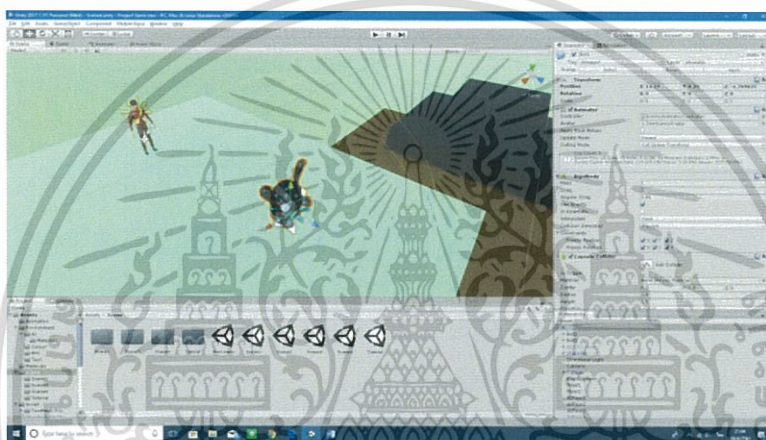
รูป 3.6 ภูมิศาสตร์ภายในโปรแกรมยูนิตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) Transform คือส่วนประกอบที่ใช้ในการกำหนดตำแหน่ง องศา และขนาดของวัตถุ
- 2) Collider คือส่วนประกอบที่ใช้ในการกำหนดว่าวัตถุที่ทับซ้อนทับ Collider นั้นคือส่วนที่สามารถจับต้องได้ Collider จะมีหลายรูปแบบ ผู้พัฒนาเกมจะต้องใช้รูปแบบให้เหมาะสมกับวัตถุของตนเอง
- 3) Script ต่าง ๆ เพื่อใช้ในการควบคุมต่าง ๆ

### 3.3.5 ศัตรูภายในเกม

การออกแบบศัตรูภายในเกมนั้น จะมีส่วนประกอบที่สำคัญสำหรับศัตรูภายในเกม ดังนี้



รูป 3.7 ศัตรูภายในโปรแกรมยูนิตี้

- 1) Transform คือส่วนประกอบที่ใช้ในการกำหนดตำแหน่ง องศา และขนาดของวัตถุ
- 2) Animator คือส่วนประกอบที่ใช้ในการทำให้วัตถุนั้นสามารถทำอนิเมชันได้
- 3) Rigidbody คือส่วนประกอบที่ทำให้วัตถุนั้นมีระบบฟิสิกส์เป็นของตนเอง
- 4) Capsule Collider คือส่วนประกอบที่ใช้ในการกำหนดว่าวัตถุที่ทับซ้อนทับ Collider นั้นคือส่วนที่สามารถจับต้องได้ Collider จะมีหลายรูปแบบ ผู้พัฒนาเกมจะต้องใช้รูปแบบให้เหมาะสมกับวัตถุของตนเอง โดยในที่นี้ใช้ในรูปแบบ Capsule
- 5) Script Player Movements เป็น Script สำหรับใช้ในการควบคุมการเคลื่อนที่ของตัวละครและการทำแอคชันต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การควบคุมภายในเกม

ระบบควบคุมของเกมจะแบ่งเป็น 2 ส่วนที่สำคัญ ดังนี้

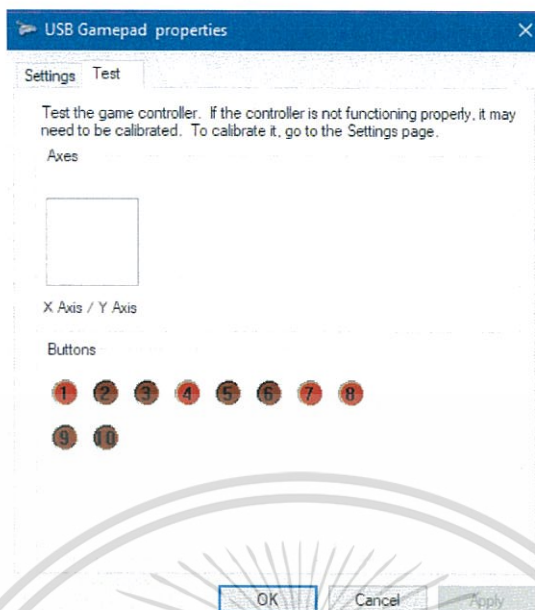
#### 3.4.1 ระบบควบคุมการเดิน

การควบคุมการเดินของตัวละครภายในเกมนั้นจะใช้อินพุทจากแดนซ์แพดโดยในอุปกรณ์แดนซ์แพดนั้นจะมีปุ่มอยู่ 11 ปุ่ม แต่ปุ่มที่เราจะใช้คือ 4 ปุ่ม คือ ขึ้น, ลง, ซ้าย และขวา เพื่อใช้ในการควบคุมตัวละครให้เดินได้แบบ 3 มิติ

#### โปรแกรม 3.2 การเขียน Script การเดินของตัวละคร

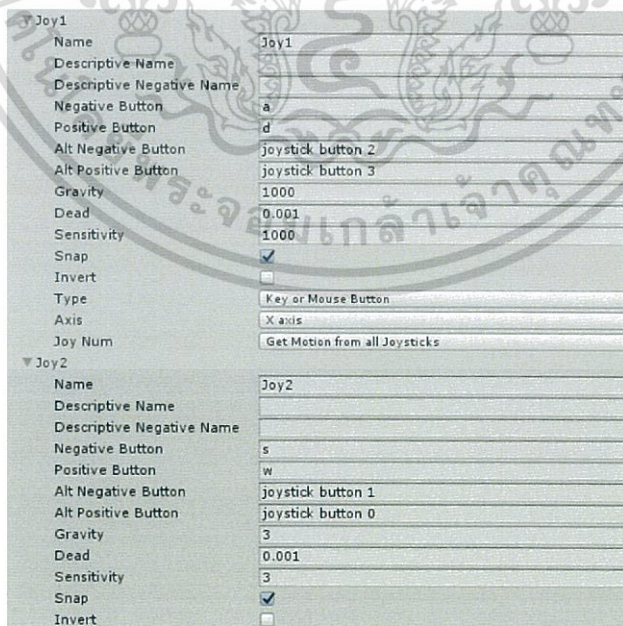
```
float h = Input.GetAxisRaw("DancePadHorizontal");
float v = Input.GetAxisRaw("DancePadVertical");
void Move(float h, float v) {
    movement.Set(h, 0, v);
    movement = movement.normalized*speed*Time.deltaTime;
    playerRigidbody.MovePosition(transform.position +
movement);
} //โดยที่ h และ v รับค่าจากการกดปุ่มของแดนซ์แพด
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.8 อินพุทของแดนซ์แพด

โดยโปรแกรมยูนิตนั้นจะสามารถรับรู้ถึงอินพุทที่เชื่อมต่อกับคอมพิวเตอร์ได้อัตโนมัติ หากเราลงไดรเวอร์ของอุปกรณ์ชนิดนั้นไว้ เราก็แค่นำอินพุทที่ตรงกับ Buttons ในรูป 3.6 ไปใส่เป็นอินพุทภายในโปรแกรมยูนิต

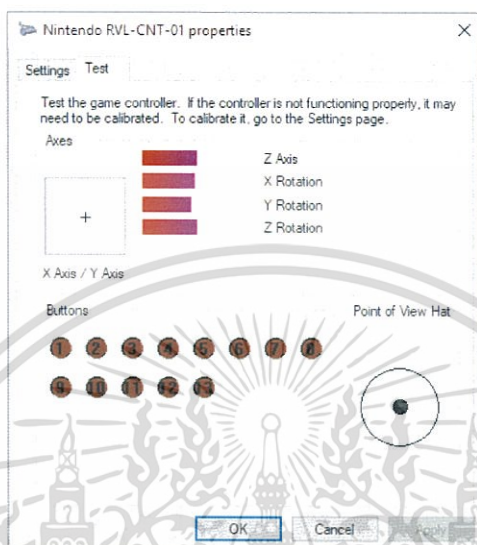


รูป 3.9 การปรับแต่งอินพุทภายในโปรแกรมยูนิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

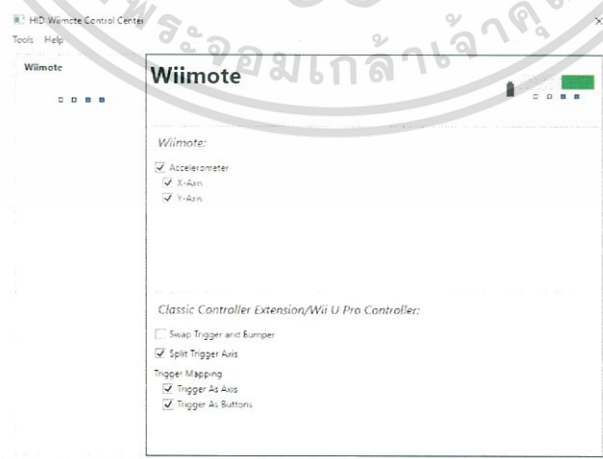
### 3.4.2 ระบบการทำแอสซันและการปรับทิศทางของตัวละคร

การทำแอสซันของตัวละครและหมุนทิศทางของตัวละครภายในเกมนั้นจะใช้อินพุทจากวีรีโมท โดยในวีรีโมทจะมีเซนเซอร์รับการเคลื่อนไหว เพื่อใช้ในการควบคุมทิศทางของตัวละครอยู่



รูป 3.10 อินพุทของวีรีโมท

โดยโปรแกรมยูนิตนั้นจะสามารถรับรู้ถึงอินพุท ที่เชื่อมต่อกับคอมพิวเตอร์ได้อัตโนมัติ หากเราลงไดรเวอร์ของอุปกรณ์ชนิดนั้นไว้ แต่ในส่วนของวีรีโมทจะต้องใช้ซอฟต์แวร์ช่วยให้คอมพิวเตอร์สามารถมองเห็นได้



รูป 3.11 โปรแกรมสำหรับเชื่อมต่อคอมพิวเตอร์และวีรีโมท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### โปรแกรม 3.3 การปรับทิศทางของตัวละคร

```

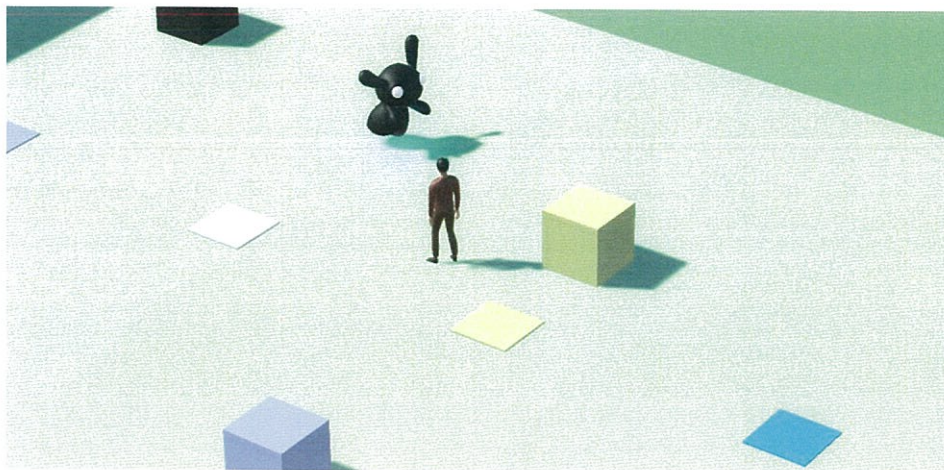
void Turning() {
    Ray camRay =
    Camera.main.ScreenPointToRay(Input.WiiRemotePosition);
    RaycastHit floorHit;
    If (Physics.Raycast(camRay, out floorHit, camRayLength,
    floorMask)) {
        Vector3 playerToWiiRemote = floorHit.point -
        transform.position;
        playerToWiiRemote.y =0f;
        Quaternion newRotation =
        Quaternion.Looktation(playerToWiiRemote);
        playerRigidbody.MoveRotation(newRotation);
    }
} // ใส่อินพุทเป็นตำแหน่งของวีรีโมทเพื่อใช้ในการควบคุมทิศทาง

```

## 3.5 การออกกำลังกายภายในเกม

### 3.5.1 การออกกำลังกายกับปริศนาภายในเกม

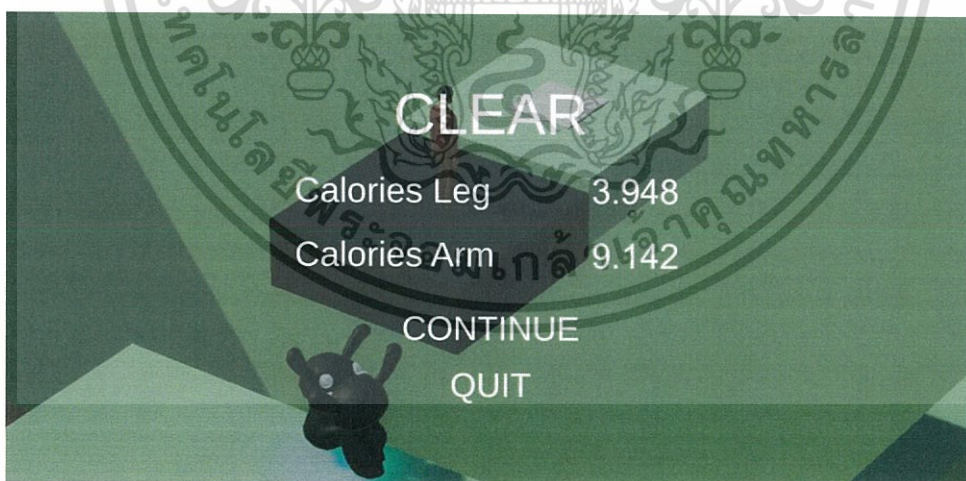
ในส่วนของ การออกกำลังกายนั้นจะสัมพันธ์กับอินพุทที่ใช้ในการเล่นเกมนั้น ซึ่งทำให้ในการเล่นเกมนั้นผู้เล่นจะต้องทำการขยับแขนและขาเพื่อใช้ในการบังคับตัวละคร ซึ่งหมายความว่าผู้เล่นจะได้ ออกกำลังกายไปพร้อมกับเล่นเกมไปด้วยนั่นเอง ภายในเกมจะมีปริศนาที่แตกต่างกันไป โดยการออกแบบปริศนาแต่ละปริศนานั้น ทางเราจะออกแบบให้ผู้เล่นจำเป็นต้องขยับแขนหรือขยับขาต่างๆ กันไปในแต่ละด่าน ตัวอย่างเช่น ผู้เล่นจะต้องตามสวิตช์เพื่อใช้เปิดเส้นทางไปยังปริศนาต่อไป โดยระหว่างทางจะมีศัตรูคอยก่อกวน



รูป 3.12 ปริศนภายในเกม

### 3.5.2 การคำนวณพลังงานที่ใช้ในการเล่นเกม

ในการเล่นเกมแต่ละครั้งผู้เล่นจะต้องทำการขยับส่วนต่างๆ ของร่างกายเพื่อใช้ในการเล่นเกม และเมื่อผู้เล่นต้องการที่จะออกจากเกม ระบบจะแสดงข้อมูลว่าในการเล่นวันนี้ผู้เล่นได้ขยับร่างกายไปอย่างไรบ้าง และคิดเป็นกิโลแคลอรีได้เท่าไร โดยจะสร้าง Script ขึ้นมาเพื่อเก็บจำนวนครั้งของการกดปุ่มที่เดคซ์แพด และการขยับแขนที่วีรีโมท จากนั้นนำข้อมูลนั้นมาคิดเป็นพลังงานใช้ไปแบบคร่าว ๆ



รูป 3.13 แคลอรีภายในเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

ในการทดลองและผลการทดลองที่ได้จากการศึกษาการใช้งานของโปรแกรม และการใช้อุปกรณ์อินพุท เพื่อนำมาพัฒนาเกมสำหรับการออกกำลังกายนั้น ในส่วนแรกของการทำโครงการ ทางผู้พัฒนาได้ศึกษาการใช้งานของโปรแกรมยูนิตี้และเบลนเดอร์มาพอสมควร และได้ทดลองพัฒนาเกมออกมาในบางส่วน และได้ทดลองใช้อินพุททั้ง 2 ชนิดรวมกับการพัฒนาเกม ซึ่งก็คือคีย์แพดและวีรีโมท

#### 4.1 การทดลองใช้ซอฟต์แวร์พัฒนาเกม

##### 4.1.1 การใช้งานโปรแกรมเบลนเดอร์

ในการทดลองใช้โปรแกรมเบลนเดอร์นั้น ทางผู้พัฒนาได้ทดลองใช้โปรแกรมในการสร้าง วัตถุและภูมิศาสตร์ภายในเกมขึ้นมา เพื่อนำไปเป็นใช้เป็นวัตถุในโปรแกรมยูนิตี้ต่อไป

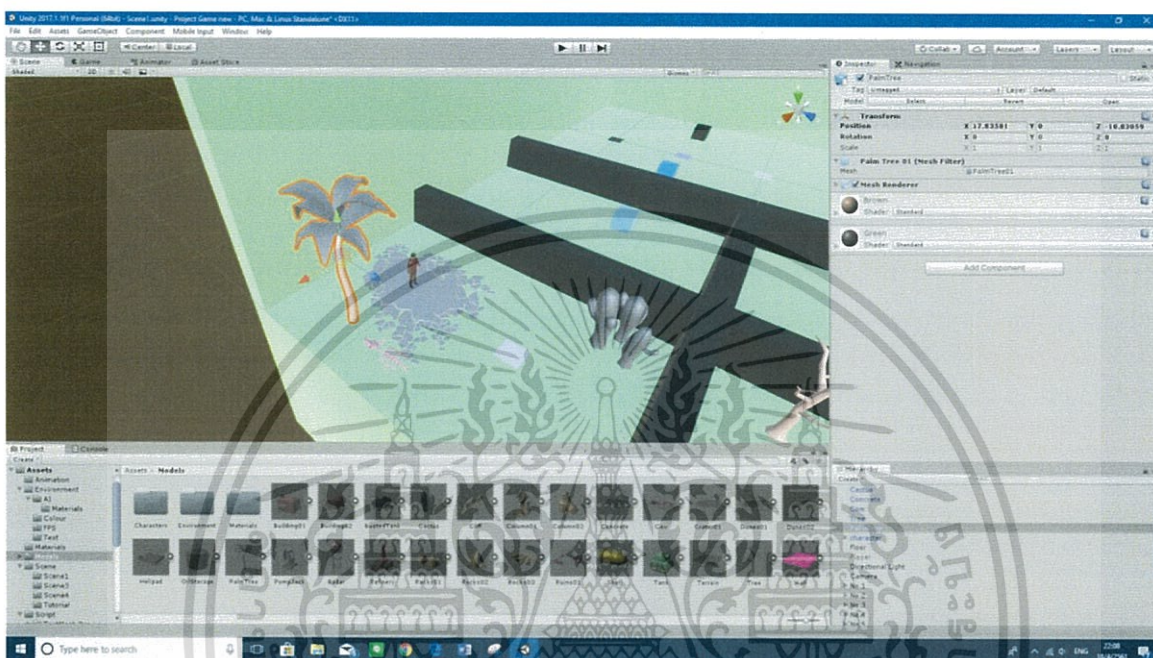


รูป 4.1 สร้างวัตถุและภูมิศาสตร์ภายในเกมด้วยโปรแกรมเบลนเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.1.2 การใช้งานโปรแกรมยูนิตี้

เมื่อสร้างวัตถุที่ต้องการใช้ภายในเกมด้วยโปรแกรมเบลนเดอร์แล้ว ทางผู้พัฒนาได้นำวัตถุเหล่านั้นมาใส่ในซีนของโปรแกรมยูนิตี้และทำการใส่ส่วนประกอบต่าง ๆ เข้าไปภายในวัตถุนั้น ๆ



รูป 4.2 การนำวัตถุและภูมิศาสตร์สร้างเป็นแผนที่ในโปรแกรมยูนิตี้

## 4.2 การเขียนโปรแกรมด้วยภาษาซีชาร์ป

### 4.2.1 การเดินของตัวละคร

ในการเดินของตัวละครในเกมนั้นจะรับค่ามาจากแกนซ์แพด โดยที่ผู้เล่นสามารถควบคุมทิศทางการเดินของตัวละครได้ 4 ทิศทาง คือ ขึ้น, ลง, ซ้าย, และขวา โดยอินพุทจากแกนซ์แพดนั้น ทางผู้พัฒนาได้นำไปเชื่อมกับอินพุทภายในโปรแกรมยูนิตี้ จากนั้นนำอินพุทภายในโปรแกรมยูนิตี้ไปเป็นอินพุทของภาษาซีชาร์ป เพื่อใช้ควบคุมการเดินของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)

(ข)

(ค)

รูป 4.3 การเดินภายในเกม

- ก) ตัวละครอยู่นิ่ง
- ข) ตัวละครเดินไปทางขวา
- ค) ตัวละครเดินขึ้น ไปทางด้านบน

#### 4.2.4 การปรับทิศทางของตัวละคร

ในการปรับทิศทางของตัวละคร ในเกมนั้นจะรับค่ามาจากวีรีโมท โดยที่ผู้เล่นสามารถควบคุมทิศทางของตัวละครได้ โดยการขยับวีรีโมทไปทางซ้ายหรือทางขวา เพื่อให้ตัวละครหมุนตาม



(ก)

(ข)

(ค)

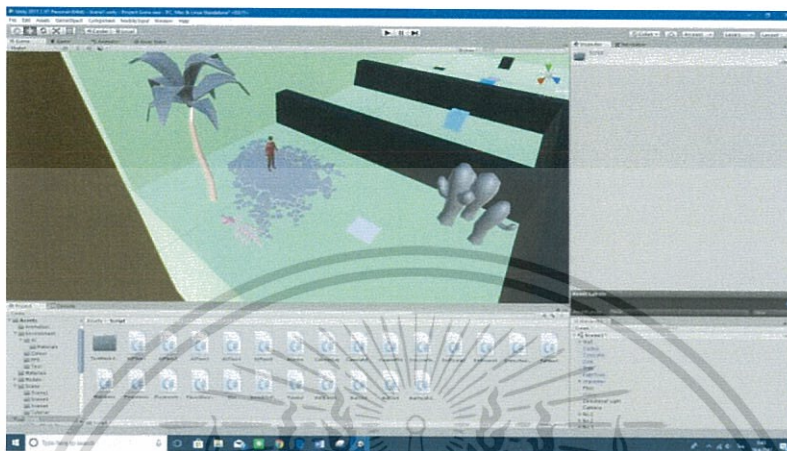
รูป 4.4 การปรับทิศทางตัวละครภายในเกม

- ก) หันทิศทางไปทางขวา
- ข) หันทิศทางไปด้านล่าง
- ค) หันทิศทางไปทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.3 การเคลื่อนที่ของกล้องจับภาพตัวละคร

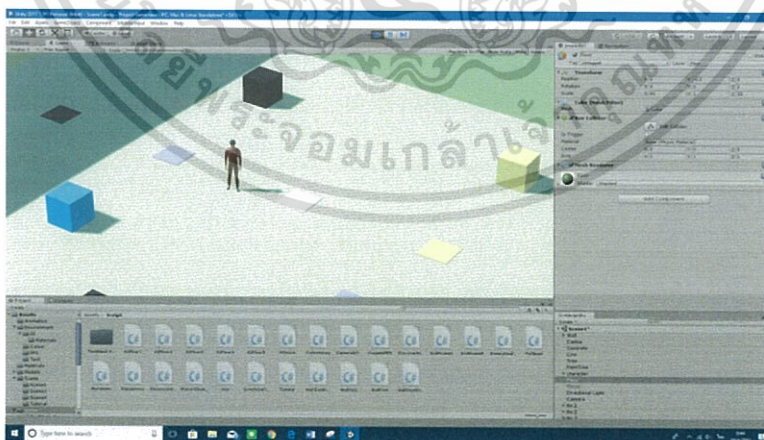
เพื่อให้ในการเล่นเกมนั้นตัวละครจะอยู่ในมุมมองให้เห็นตลอดเวลา นั้น ทางผู้พัฒนาได้ทำการเขียนโปรแกรมภาษาซีชาร์ป เพื่อกำหนดตำแหน่งของกล้องให้ติดตามตัวละครไปตลอดเวลา



รูป 4.5 การจับภาพของกล้อง

### 4.2.4 การเคลื่อนกล้องเพื่อเปิดประตูไปยังด่านต่อไป

เพื่อเป็นการทดสอบในการพัฒนาเกม ทางผู้พัฒนาเกมจึงได้พัฒนาปริศนาขึ้นมาเพื่อใช้ในการทดสอบเล่น และเมื่อเล่นจบจะเปิดประตูไปยังด่านต่อ ๆ ไปได้

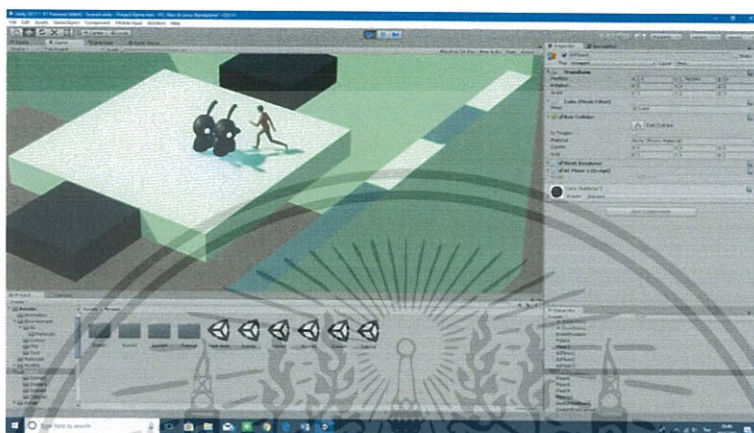


รูป 4.6 ด่านที่ผู้เล่นต้องผลักกล่องไปทับสวิตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.5 การสร้างศัตรูขึ้นในเกม

ศัตรูที่เกิดขึ้นภายในเกมนั้น จะโผล่มาเพื่อให้ผู้เล่นเล่นเกมได้ยากขึ้น โดยศัตรูนั้นจะเคลื่อนไปคอยก่อกวนผู้เล่นอยู่ตลอดเวลา โดยจะตรวจจับผู้เล่นจากตำแหน่งที่ผู้เล่นยืนอยู่



รูป 4.7 ศัตรูที่เกิดขึ้นมาก่อกวนผู้เล่นภายในเกม

#### 4.3 การออกกำลังกายระหว่างเล่นเกม

เราได้ทำการทดลองว่าเราสามารถออกกำลังกายได้มากเท่าไรในการเล่นเกมนาน 30 นาที โดยทำการทดสอบ 10 ครั้ง

##### ตารางที่ 4.1 ผลการทดลองการออกกำลังกายระหว่างเล่นเกม 30 นาที

No.	Arm	Leg	Total
1	21.2	124	145.2
2	19.8	129	148.8
3	20.3	131	151.3
4	18.5	91	109.5
5	19.4	116	135.4
6	19.9	120	139.9
7	20.3	105	125.3
8	22.3	98	110.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9	20.1	121	141.1
10	18.1	90	108.1
Mean	19.9	112.5	134.4

จากการทดลองการออกกำลังกายในระหว่างการเล่นเกมเป็นเวลา 30 นาที ได้ค่าเฉลี่ยของแคลอรีที่เผาผลาญไปในขณะเล่นเกมเป็น 134.4 แคลอรี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# บทสรุปและข้อเสนอแนะ

### 5.1 บทสรุป

โครงการพัฒนาเกมสำหรับการออกกำลังกายนั้น สามารถทำให้ผู้เล่นได้ขยับร่างกายในส่วนต่างๆได้ไม่ว่าจะเป็นแขนหรือขา จึงทำให้ผู้เล่นได้ออกกำลังกายจากการเล่นเกมที่มีการคำนวณให้ผู้เล่นได้เห็นเป็นตัวเลขวหลังจากเล่นเกมจบด้าน โดยผู้จัดทำสามารถทำตัวเกมออกมาโดยใช้โปรแกรมยูนิตีในการทำระบบเกม ซึ่งทำการโปรแกรมระบบโดยใช้ภาษาซีชาร์ป และออกแบบตัวละครโดยใช้โปรแกรมเบลนเดอร์ได้ ในส่วนของการเชื่อมต่อระหว่างซอฟต์แวร์และอินพุทนั้น สามารถควบคุมตัวละครภายในเกมได้โดยการควบคุมขาหรือเค็ลอื่นที่ตัวละครใช้ Dance pad ในการควบคุม และการควบคุมแขนหรือหมุนตัวละครทำได้โดยการหมุนวีรี โมท

### 5.2 ปัญหาและอุปสรรค

ในการศึกษาข้อมูลเบื้องต้นของการทำโครงการนั้นค่อนข้างจะเป็นไปอย่างล่าช้า สาเหตุเนื่องมาจากตัวซอฟต์แวร์แต่ละชนิด ถือเป็นเรื่องใหม่สำหรับผู้จัดทำโครงการ ทำให้การศึกษาด้วยตนเองนั้นค่อนข้างยาก รวมถึงอุปกรณ์ที่ใช้เป็นอินพุททั้ง 2 อุปกรณ์ด้วย ซึ่งก็คือ แคนซ์แพด และวีรี โมท อุปกรณ์ทั้งสองก็ถือเป็นอินพุทใหม่ที่ทางผู้จัดทำโครงการยังไม่เคยได้สัมผัสมาก่อน โดยเฉพาะวีรี โมทที่ไม่สามารถต่อเข้ากับคอมพิวเตอร์ได้โดยตรง แคมคอมพิวเตอร์บางเครื่องนั้นยังไม่สามารถต่อได้แบบสมบูรณ์ด้วย เวลาที่ใช้ในส่วนของการพัฒนาเกมนั้นก็ถือว่ากินเวลาไปพอสมควร ทำให้การศึกษาเรื่องการออกกำลังกายผ่านเกมเพื่อนำมาใส่เป็นปริศนาภายในเกมนั้นเหลือนเวลาน้อยจนศึกษาได้ไม่ครอบคลุมสิ่งที่ต้องการในตอนนี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 แนวทางในการพัฒนาต่อ

- 1) เปลี่ยนอินพุทที่ใช้ เนื่องด้วยตัวอินพุทที่ผู้จัดทำโครงการใช้นั้นค่อนข้างจะใช้ยาก และไม่ได้ถูกออกแบบมาใช้พัฒนาต่อได้ และยังมีข้อจำกัดมากสำหรับการเคลื่อนไหวของร่างกาย
- 2) ประยุกต์เกมกับการออกกำลังกายให้มากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

Unity **Learn with Unity**. [Online].

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

Unity **Public-Relations**. [Online].

Available : <https://unity3d.com/public-relations>

Blender **Blender About us**. [Online].

Available : <https://www.blender.org/about/>

Unity **Documentation for Unity**. [Online].

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

Julian Lohr **HID Wiimote – A Windows Device Driver For Wii Remote**. [Online].

Available : <https://www.julianloehr.de/educational-work/hid-wiimote/>

Microsoft **Hardware Dev Center Plug and Play Manager**. [Online].

Available : <https://docs.microsoft.com/th-th/windows-hardware/drivers/install/pnp-manager>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้