

การสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ
Smart Jump Rope Game and Application

โรงเรียนอุทัยวิทยาคม
จังหวัดอุทัยธานี

คณะผู้จัดทำ

1. นางสาวปวีณ์ลดา จิงสมเจตไพศาล
2. นางสาวพิมพ์กานต์ คำจันทร์
3. นางสาวภิญญดา ชนะดี

รายงานนี้เป็นส่วนหนึ่งของสาขาวิชาคอมพิวเตอร์
การนำเสนอโครงงานของนักเรียนจากโรงเรียนโครงการห้องเรียนพิเศษวิทยาศาสตร์ คณิตศาสตร์
เทคโนโลยีและสิ่งแวดล้อม เครือข่ายภาคกลางตอนบน ประจำปีการศึกษา 2567

การสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ
Smart Jump Rope Game and Application

โรงเรียนอุทัยวิทยาคม
จังหวัดอุทัยธานี

คณะผู้จัดทำ

1. นางสาวปวีณ์ลดา จิงสมเจตไพศาล
2. นางสาวพิมพ์กานต์ คำจันทร์
3. นางสาวภิญญดา ชนะดี

ครูที่ปรึกษา

1. นายถิรพัฒน์ วิเชียรรัตน์
2. นายฉัตรชัย นาคทอง

รายงานนี้เป็นส่วนหนึ่งของสาขาวิชาคอมพิวเตอร์
การนำเสนอโครงงานของนักเรียนจากโรงเรียนโครงการห้องเรียนพิเศษวิทยาศาสตร์ คณิตศาสตร์
เทคโนโลยีและสิ่งแวดล้อม เครือข่ายภาคกลางตอนบน ประจำปีการศึกษา 2567

บทคัดย่อ

โครงการนี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อ 1. สร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะจากการเขียนคำสั่งโปรแกรมผ่านเซนเซอร์ตรวจจับและวัดค่ามุม MPU-6050 2. ทดสอบประสิทธิภาพของอุปกรณ์ตรวจจับการกระโดด และ 3. สำรวจความพึงพอใจสำหรับผู้ใช้งานแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ มีกลุ่มตัวอย่างเป็นนักเรียนในโรงเรียนอุทัยวิทยาคม จำนวน 100 คน โดยทำการเชื่อมบอร์ด Arduino ที่ต่อกับแบตเตอรี่ 9V เชื่อมต่อข้อมูลกับเกมกระโดดเชือกอัจฉริยะ แล้วกระโดดเก็บเหรียญที่แสดงบนหน้าจอ เพื่อเผาผลาญแคลอรีให้ได้เท่ากับจำนวนแคลอรีที่ต้องเผาผลาญต่อวันตามที่คำนวณไว้ในแอปพลิเคชัน

จากการศึกษาพบว่าแอปพลิเคชันคำนวณแคลอรีสามารถคำนวณพลังงานที่ควรเผาผลาญและพลังงานที่แนะนำต่อวันได้ และเมื่อเขียนโปรแกรมตรวจจับในบอร์ด Arduino เขียนโปรแกรม C# ใน Unity ตรงส่วนของ Player และเรียกใช้คำสั่งตรวจจับการเคลื่อนไหวในบอร์ด Arduino ให้ส่งค่าจากคำสั่งโปรแกรมตรวจจับค่ามุมในหน่วยเรเดียน เพื่อสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ เมื่อจ่อเข้ามามากกว่า 1 เรเดียนและเมื่อน้อยกว่า -1 เรเดียนกับพื้นโลกอุปกรณ์วัดค่ามุมการกระโดดสามารถตรวจจับการจ่อเข้าได้ โดยหากบอร์ด Arduino ส่งค่าเป็น 1 ตัวละครจะกระโดดเพื่อเก็บเหรียญและนับค่าแคลอรี ส่งค่าเป็น 2 ตัวละครจะเลิกกระโดด และส่งค่าเป็น 3 ตัวละครจะยืนนิ่ง โดยอุปกรณ์ตรวจจับการกระโดดมีความแม่นยำในการตรวจจับอยู่ที่ร้อยละ 90.00 ซึ่งแม่นยำในระดับมากที่สุด อีกทั้งความพึงพอใจเรื่องการให้ความสนุกสนานของเกม ประสิทธิภาพการตรวจจับของเซนเซอร์ ความสวยงามของหน้าตาเกม และความพึงพอใจโดยรวมต่อเกมอยู่ในระดับความพึงพอใจมากที่สุด (ค่าเฉลี่ย = 4.66 และส่วนเบี่ยงเบนมาตรฐาน = 0.623123)

กิตติกรรมประกาศ

โครงการเรื่องการสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะสามารถดำเนินการจนประสบความสำเร็จล่วงไปได้ด้วยดีเนื่องจากได้รับความกรุณาเป็นอย่างสูงจาก คุณครูธิพัฒน์ วิเชียรรัตน์ และคุณครูฉัตรชัย นาคทอง ที่ปรึกษาโครงการที่ช่วยแนะนำและตรวจสอบแก้ไขข้อบกพร่องต่างๆด้วยความใส่ใจอย่างยิ่งจนคณะผู้จัดทำสามารถดำเนินงานได้สำเร็จล่วง

ขอขอบพระคุณ คุณครูวีรญาดา บางแบ่ง ที่ปรึกษาโครงการพิเศษ ที่ได้ตรวจทาน ปรับปรุง และแนะแนวทางการจัดทำโครงการ

ขอขอบพระคุณ นายยุทธกานต์ สาจัญ และ นายไพสิฐ เลิศอนันต์พิพัฒน์ รุ่นพี่ผู้มีประสบการณ์ในการทำโครงการคอมพิวเตอร์ที่คอยให้ปรึกษาตลอดระยะเวลาในการทำโครงการ

ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ที่สนับสนุนทุนในการทำโครงการครั้งนี้รวมทั้งเป็นกำลังใจและให้โอกาสการศึกษาอันมีค่าต่อตัวผู้จัดทำอย่างยิ่ง

สุดท้ายนี้ขอขอบคุณเพื่อน ๆ ที่คอยช่วยเหลือในการทำโครงการและคอยให้คำปรึกษาต่าง ๆ ที่เกี่ยวกับโครงการนี้รวมถึงคอยสนับสนุนคณะผู้จัดทำ

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	จ
สารบัญรูปภาพ	ฉ
บทที่ 1 บทนำ	
ที่มาและความสำคัญ	1
วัตถุประสงค์	1
นิยามศัพท์เฉพาะ	2
ขอบเขตการศึกษาโครงการ	2
ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 เอกสารที่เกี่ยวข้อง	
Unity	3
การออกกำลังกายโดยการกระโดด	4
ค่า BMR และ TDEE	4
โปรแกรมออกแบบ	5
Arduino IDE	5
Gyroscope mpu-6050	6
App Inventor	6
งานวิจัยที่เกี่ยวข้อง	6
บทที่ 3 อุปกรณ์และวิธีดำเนินการทดลอง	
วัสดุ อุปกรณ์ และโปรแกรมที่ใช้พัฒนา	8
ขั้นตอนการดำเนินงาน	8
บทที่ 4 ผลการทดลอง	
การสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ	17
การทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดด	17
การประเมินความพึงพอใจของผู้ใช้งานแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ	18
บทที่ 5 สรุปและอภิปรายผลการทดลอง	
สรุปผลการทดลอง	19
อภิปรายผลการทดลอง	19
ข้อเสนอแนะ	20

สารบัญ (ต่อ)

เรื่อง	หน้า
บรรณานุกรม	21
ภาคผนวก	22

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 4.1 ผลการสำรวจความพึงพอใจที่มีต่อโครงการจากกลุ่มผู้ใช้งาน	18

สารบัญรูปภาพ

ภาพที่	หน้า
ภาพที่ 2.1 ลักษณะของ Gyro Sensor	6
ภาพที่ 3.1 Arduino IDE ก่อน void setup	9
ภาพที่ 3.2 Arduino IDE void set up	10
ภาพที่ 3.3 Arduino IDE void loop	11
ภาพที่ 3.4 คำสั่งโปรแกรม C# สำหรับออกเกม เปลี่ยนหน้าเกม และ รีเซ็ตเกม	12
ภาพที่ 3.5 คำสั่งโปรแกรม C# Player (ตัวละคร)	14
ภาพที่ 3.6 คำสั่งโปรแกรม C# การนับเหรียญและการ Restart ค่าเคลอรี่	14

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปีที่ผ่านมา กรมควบคุมโรคได้รายงานการระบาดของโรคติดต่อสำคัญในประเทศไทย โดยเฉพาะโรคไข้เลือดออก โรคไข้หวัดใหญ่ และโรคโควิด-19 ซึ่งพบการระบาดอย่างมาก โรคไข้เลือดออก มีจำนวนผู้ป่วยสูงถึง 147,412 ราย หรืออัตราการป่วย 222.91 ต่อประชากรแสนคน ซึ่งเพิ่มขึ้นถึง 3.4 เท่าจากปีที่ผ่านมา และมีผู้เสียชีวิต 174 ราย โรคไข้หวัดใหญ่พบผู้ป่วย 460,325 รายในปีเดียวกัน พร้อมมีผู้เสียชีวิต 29 ราย ส่วนโรคโควิด-19 แม้การระบาดจะลดลง แต่ยังคงมีผู้ป่วยในโรงพยาบาล 37,863 ราย และมีผู้เสียชีวิต 845 ราย โดยส่วนใหญ่เป็นผู้สูงอายุและผู้ที่มีโรคประจำตัว อัตราการป่วยที่สูงในหลายโรคดังกล่าวเป็นสัญญาณเตือนถึงปัญหาสุขภาพที่ต้องได้รับการแก้ไขอย่างเร่งด่วน หากไม่ได้รับการดูแลและป้องกันที่เหมาะสม โรคเหล่านี้สามารถแพร่ระบาดในวงกว้างและก่อให้เกิดผลกระทบอย่างรุนแรงต่อระบบสาธารณสุข เศรษฐกิจ และคุณภาพชีวิตของประชาชน เช่น การจัดสรรทรัพยากรเพื่อควบคุมการระบาด การให้บริการสุขภาพที่ล่าช้าเนื่องจากผู้ป่วยมีจำนวนมาก และความเครียดที่เกิดจากการดูแลรักษาผู้ป่วย

การออกกำลังกายเป็นหนึ่งในวิธีที่มีประสิทธิภาพในการลดความเสี่ยงของการเกิดโรคต่าง ๆ โดยช่วยเสริมสร้างระบบภูมิคุ้มกันให้แข็งแรง ทำให้ร่างกายสามารถต่อสู้กับเชื้อโรคได้ดีขึ้น อีกทั้งยังช่วยลดความเสี่ยงของภาวะซึมเศร้า ความวิตกกังวล และโรคไม่ติดต่อหลายชนิด เช่น โรคหัวใจ เบาหวาน และความดันโลหิตสูง การออกกำลังกายอย่างสม่ำเสมอจึงเป็นสิ่งสำคัญในการดูแลสุขภาพ

การกระโดดเชือกเป็นการออกกำลังกายที่มีประโยชน์สูง โดยจะเพิ่มอัตราการเต้นของหัวใจ ลดความเสี่ยงของการเกิดโรคต่าง ๆ และช่วยเผาผลาญแคลอรีได้เร็วเมื่อเทียบกับการออกกำลังกายแบบอื่น ๆ นอกจากนี้ยังพัฒนาเรื่องการประสานงานและการทรงตัวในร่างกาย รวมถึงเสริมสร้างความหนาแน่นของกระดูก และเพิ่มระดับเซโรโทนินซึ่งช่วยบรรเทาความเครียดและอาการซึมเศร้า

จากข้อมูลดังกล่าว ผู้จัดทำเล็งเห็นความสำคัญของการออกกำลังกาย จึงพัฒนาแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะผ่านโปรแกรม Unity จากการเขียนคำสั่งโปรแกรมผ่านเซนเซอร์ MPU-6050 ซึ่งได้นำเทคโนโลยีเกมมาประยุกต์ใช้กับการออกกำลังกายอย่างการกระโดดเชือก ซึ่งจะช่วยเสริมสร้างความสนุกสนานในการออกกำลังกายและเป็นแนวทางใหม่ในการส่งเสริมสุขภาพให้กับประชาชน

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะจากการเขียนคำสั่งโปรแกรมผ่านเซนเซอร์ตรวจจับและวัดค่ามุม MPU-6050

1.2.2 เพื่อทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดด

1.2.3 เพื่อสำรวจความพึงพอใจของผู้ใช้งานแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ

1.3 นิยามศัพท์เฉพาะ

Unity	หมายถึง โปรแกรมสำหรับพัฒนาตัวเกมกระโดดเชือกอัจฉริยะ
Arduino IDE	หมายถึง โปรแกรมที่ใช้สำหรับการเขียนโปรแกรมควบคุม และทำการแปลงไฟล์ดังกล่าวเพื่อนำไปอัปโหลดลง ยังบอร์ด Arduino
App Inventor	หมายถึง เครื่องมือที่ใช้สร้างแอปพลิเคชันคำนวณแคลอรี
เซนเซอร์ตรวจจับและวัดค่ามุม	หมายถึง เซนเซอร์วัดความเร่งและการหมุนรอบ MPU- 6050 ซึ่งในโครงงานนี้ใช้สำหรับตรวจจับและวัดค่ามุมการงอเข้า

1.4 ขอบเขตของโครงการ

- 1.4.1 โปรแกรม Unity เป็นโปรแกรมที่ใช้ในการพัฒนาตัวเกมกระโดดเชือก
 - 1.4.1.1 ศึกษาความแม่นยำของการรับค่าเมื่อผู้ใช้งานทดสอบ
 - 1.4.1.2 ศึกษาการสร้าง Animation ให้กับตัวละครในเกม
 - 1.4.1.3 ศึกษา component ในเกม เพื่อสร้างฟังก์ชันให้กับส่วนประกอบภายในเกม
 - 1.4.1.4 ศึกษาการเขียนคำสั่งโปรแกรมภาษา C# เพื่อเขียนคำสั่งการเคลื่อนไหวภายในเกม
 - 1.4.1.5 ศึกษาโปรแกรม visual studio เพื่อเปลี่ยนคำสั่งภาษา C# เป็นการเคลื่อนไหวในตัวละคร
- 1.4.2 โปรแกรม Arduino IDE เป็นโปรแกรมสำหรับเขียนคำสั่งลงในบอร์ดควบคุม
 - 1.4.2.1 ศึกษาวิธีการวัดค่าจากเซนเซอร์ตรวจจับและวัดค่ามุม MPU-6050
 - 1.4.2.2 ศึกษาการเขียนคำสั่งโปรแกรมในโปรแกรม Arduino IDE
- 1.4.3 โปรแกรม App Inventor เป็นโปรแกรมสำหรับพัฒนาแอปพลิเคชันคำนวณแคลอรี
 - 1.4.3.1 ศึกษาการเขียนคำสั่งโปรแกรมในโปรแกรม App Inventor เพื่อสร้างแอปพลิเคชันคำนวณแคลอรี
 - 1.4.3.2 ศึกษาองค์ประกอบต่าง ๆ เช่น layout button ฯลฯ เพื่อใช้ในการสร้างแอปพลิเคชัน
- 1.4.4 ประชากรและกลุ่มตัวอย่าง
 - นักเรียนของโรงเรียนอุทัยวิทยาคม จำนวน 100 คน
- 1.4.5 สถานที่จัดทำโครงการ
 - โรงเรียนอุทัยวิทยาคม
- 1.4.6 ระยะเวลาในการศึกษา
 - มิถุนายน 2566 - สิงหาคม 2567

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

- 1.5.1 ได้พัฒนาทักษะการเขียนโปรแกรม
- 1.5.2 ทำให้เกิดทักษะกระบวนการทางวิทยาศาสตร์และเทคโนโลยี
- 1.5.3 ได้ส่งเสริมการออกกำลังกาย และเพิ่มแนวโน้มการเคลื่อนไหวทางกายของประชากร
- 1.5.4 เพิ่มความตระหนักรู้ด้านสุขภาพ และลดอัตราการเกิดโรคต่าง ๆ ลง

บทที่ 2

เอกสารที่เกี่ยวข้อง

ในการจัดทำโครงงานคอมพิวเตอร์เพื่อสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ คณะผู้จัดทำได้ศึกษาข้อมูลเอกสารที่เกี่ยวข้อง ดังนี้

2.1 Unity

2.1.1 ความหมาย

Unity คือ Game Engine หรือเครื่องมือสำหรับการพัฒนาเกมได้ทั้งรูปแบบ 2 มิติและ 3 มิติที่มีการทำงานข้าม Platform ได้ทั้งแบบ Desktop, Mobile, Web ผู้ใช้สามารถสร้างเกมให้รองรับใน Platform ดังกล่าวได้โดยการพัฒนาเกมเพียงครั้งเดียว ตัว Engine ประกอบด้วย API ในการช่วยเหลือนักพัฒนาด้านการเขียน Script โปรแกรม โดยภาษาโปรแกรมที่ใช้เขียนคือ C# ในปัจจุบัน Unity ได้ถูกนำไปใช้ในงานด้านต่าง ๆ ที่ไม่ได้เกี่ยวข้องกับการทำเกมอย่างเดียว เช่น งานด้านภาพยนตร์, สถาปัตยกรรม และวิศวกรรม รวมไปถึงการโต้ตอบกับผู้ใช้โดยการพัฒนาในรูปแบบของงานด้าน AR และ VR เป็นต้น

2.1.2 ภาษา C#

C# คือ ภาษาคอมพิวเตอร์ประเภท object-oriented programming พัฒนาโดย Microsoft โดยมีจุดมุ่งหมายในการรวมความสามารถคำนวณของ C++ ด้วยการโปรแกรมง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ Java C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคืออำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึง Object ของโปรแกรมหรือเมธอด โดยปราศจากความต้องการให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่แทนที่การคัดลอกซ้ำ C# ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi , ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP (โปรแกรมเชิงวัตถุ) มากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีสิ่งที่เกินความจำเป็นน้อยลง C# ถูกรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO และปัจจุบันไม่ใคร่ซอฟต์แวร์ยังพัฒนาภาษานี้อย่างต่อเนื่อง (ปัจจุบันเป็นเวอร์ชัน 3.0)

2.1.3 Visual Studio

Visual Studio เป็นโปรแกรมที่ใช้สร้างเว็บเพจเสมือนจริง หรือใช้พัฒนา App ผ่านการเขียน Mobile App ด้วย Visual Studio เป็นตัวช่วยสำหรับผู้ที่ต้องการสร้างเว็บเพจโดยไม่ต้องเขียนภาษา HTML

Visual Studio ทำให้ผู้ใช้งานสามารถจัดวางข้อความ รูปภาพ ตาราง วิดีโอ รวมไปถึงองค์ประกอบต่าง ๆ ภายในเว็บเพจได้ตามความต้องการอย่างสวยงาม โดยที่ไม่ต้องใช้ภาษาสคริปต์ที่มีความยุ่งยากซับซ้อน และนอกจากนี้ Visual Studio ยังรองรับการเขียนโปรแกรมได้หลายภาษาอีกด้วย

2.2 การออกกำลังกายโดยการกระโดด

การกระโดด (Jumping) หมายถึง การกระโดดสปริงตัวขึ้นจากพื้นด้วยเท้าทั้งสองพร้อม ๆ กัน และลงสู่พื้นด้วยเท้าทั้งสองเช่นกัน ในการลงสู่พื้นจะต้องย่อเข่าเพื่อลดแรงกระแทกและลงด้วยปลายเท้า

2.3 ค่า BMR และ TDEE

BMR คือ Basal Metabolic Rate หรือ ค่าที่บ่งบอกถึงพลังงานที่ร่างกายต้องการน้อยที่สุดในแต่ละวัน เพื่อให้ระบบต่าง ๆ สามารถทำงานได้ แม้ในตอนที่เราอยู่เฉยๆ หรือไม่ได้ทำกิจกรรมใดๆ นั้นหมายความว่า ค่า BMR ที่เราได้นั้น คือ ปริมาณแคลอรีในหนึ่งวันที่ร่างกายเราใช้ เมื่อเราได้ค่า BMR ของเรามาแล้ว จะได้ทราบว่าเราควรกินอาหารในปริมาณเท่าใด เพื่อให้สอดคล้องกับการใช้พลังงานของร่างกายของเรา

วิธีการคำนวณค่า BMR มีหลายสูตรจึงได้เลือกหนึ่งสูตรการคำนวณที่ได้รับความนิยมอย่างแพร่หลาย ซึ่งสูตรนี้จะใช้น้ำหนัก ส่วนสูง อายุ และเพศในการคำนวณ ซึ่งในผู้ชายและผู้หญิงมีสูตรในการคำนวณที่แตกต่างกัน เพราะชายและหญิงนั้นมีอัตราการเผาผลาญพลังงานในแต่ละวันที่ไม่เท่ากัน จึงต้องแยกคำนวณตามสูตรดังต่อไปนี้

BMR สำหรับผู้ชาย = $66 + (13.7 \times \text{น้ำหนักตัวปัจจุบันเป็นกิโลกรัม}) + (5 \times \text{ส่วนสูงปัจจุบันเป็นเซนติเมตร}) - (6.8 \times \text{อายุปัจจุบัน})$

BMR สำหรับผู้หญิง = $665 + (9.6 \times \text{น้ำหนักตัวปัจจุบันเป็นกิโลกรัม}) + (1.8 \times \text{ส่วนสูงปัจจุบันเป็นเซนติเมตร}) - (4.7 \times \text{อายุปัจจุบัน})$

ซึ่งค่าที่ได้จากการคำนวณ คือ จำนวนพลังงานที่ร่างกายของแต่ละคนควรได้รับเป็นอย่างน้อยในแต่ละวัน

การคำนวณปริมาณแคลอรีที่เหมาะสมกับกิจกรรมของเราในแต่ละวัน อย่างที่ได้บอกไปแล้วว่าการได้ค่า BMR มานั้นเป็นเพียงจำนวนแคลอรีที่ร่างกายใช้โดยไม่มีกิจกรรมใดๆ แต่เมื่อเราออกกำลังกายหรือทำกิจกรรมอื่นๆ กิจกรรมเหล่านี้ส่งผลให้ร่างกายใช้พลังงานมากขึ้น ดังนั้นหากเราทำกิจกรรมอื่น ๆ ร่างกายของเราจะใช้พลังงานประมาณเท่าไรต่อวัน เพื่อให้เราสามารถวางแผนการกินเพื่อลดน้ำหนักได้อย่างถูกต้อง เรากำลังพูดถึงค่า TDEE (Total Daily Energy Expenditure) คือ ปริมาณการใช้พลังงานของร่างกายของ แต่ละคนในหนึ่งวัน ซึ่งแต่ละคนจะใช้พลังงานไม่เท่ากัน ขึ้นอยู่กับค่า BMR และประเภทของกิจกรรมของแต่ละคนด้วยเพื่อคำนวณหาแคลอรีที่ร่างกายต้องการต่อวันโดยเฉลี่ย โดยใช้ค่า BMR ที่หามาได้เป็นตัวตั้ง

นั่งทำงานอยู่กับที่ ไม่ได้ออกกำลังกายเลย = $\text{BMR} \times 1.2$

ออกกำลังกายเบาๆ 1-3 วันต่อสัปดาห์ = $\text{BMR} \times 1.375$

ออกกำลังกายปานกลาง 3-5 วันต่อสัปดาห์ = $\text{BMR} \times 1.55$

ออกกำลังกายหนัก 6-7 วันต่อสัปดาห์ = $\text{BMR} \times 1.7$

ออกกำลังกายอย่างหนักเป็นประจำ เพื่อซ่อมในการแข่งขัน = $BMR \times 1.9$

การควบคุมอาหารเพื่อลดน้ำหนัก เมื่อเราทราบค่า TDEE นั่นคือปริมาณแคลอรีที่ร่างกายของเราต้องการใช้ในการทำกิจกรรมในแต่ละวันแล้ว ในการลดน้ำหนัก แนะนำให้ลดปริมาณพลังงานให้น้อยกว่าที่ร่างกายเราใช้วันละประมาณ 500 แคลอรี แต่เมื่อลดออก 500 แคลอรีแล้ว จะต้องหามีน้อยกว่าค่า BMR ที่เราคำนวณได้ ปริมาณแคลอรีที่แนะนำเพื่อลดน้ำหนัก = $TDEE - 500$ (ปริมาณแคลอรีที่แนะนำหามีน้อยกว่า BMR) เพราะถ้าเรากินน้อยกว่าค่า BMR จะทำให้ร่างกายได้รับพลังงานน้อยเกินไป เมื่อร่างกายเรียนรู้ว่าได้รับพลังงานน้อยมากก็จะไม่ยอมใช้พลังงานตามปกติ อาจส่งผลให้ระบบเผาผลาญของเราพังได้

2.4 โปรแกรมออกแบบ

2.4.1 Canva

เป็นแอปพลิเคชันสำหรับสร้างสื่อการนำเสนอหลากหลายรูปแบบ เช่น Presentation, Poster, Card, Resume, Certificate, Infographic เป็นต้น ซึ่ง Canva นั้นจะมี Template หรือรูปแบบสำเร็จ รวมถึงขนาดมาตรฐานให้เลือกหรือ ผู้ใช้สามารถกำหนดขนาดหรือออกแบบสื่อเองจากหน้ากระดาษที่วางเปล่าเองได้ มาพร้อมกับ Template สำเร็จรูป หรือภาพประกอบแบบสำเร็จอย่างหลากหลายที่สามารถเลือกปรับแต่งได้เอง มีความสวยงาม และสามารถแบ่งปันให้แก่ผู้อื่นได้

2.4.2 Procreate

Procreate คือ แอปสำหรับทำกราฟิกและออกแบบ ที่ผู้ใช้งานสามารถวาดภาพด้วย Apple Pencil หรือ Stylus Pen ปากกาที่มีไว้ใช้กับสมาร์ทโฟนหรือแท็บเล็ต โดยแอป ข้อดีของแอป Procreate อยู่ที่ การใช้งานที่ง่าย ไม่มีเครื่องมือที่ซับซ้อนมากนัก การทำงานหลาย ๆ ส่วน ก็คล้ายกับ Photoshop เช่น การสร้าง Layer, การดูสี รวมถึงมี Brush หรือหัวแปรง สำหรับวาดภาพ หลายร้อยแบบที่มาพร้อมกับตัวแอป ซึ่งถ้า Brush ที่ให้มายังไม่ถูกใจ เราก็สามารถออกแบบ Brush ของเราเองได้อีกด้วย

2.5 Arduino IDE

Arduino IDE เป็นโปรแกรมที่ใช้งานลักษณะ Open source ซึ่ง Arduino IDE จะทำหน้าที่ติดต่อระหว่างคอมพิวเตอร์ไม่ว่าจะเป็นระบบ Windows, Mac OS X หรือ Linux กับ บอร์ด Arduino ซึ่งโปรแกรมนี้ออกแบบให้ง่ายต่อการเขียนโค้ดและอัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino Arduino IDE ส่วน IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น ส่วนในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชันอื่น ๆ ของ Arduino

2.6 Gyroscope mpu-6050

Gyroscope mpu-6050 คือ เซนเซอร์ที่มีไว้สำหรับตรวจจับลักษณะการหมุนของสมาร์ทโฟน โดยเป็นการตรวจจับแบบ 3 แกน (3-Axes) เช่นเดียวกันกับ Accelerator Sensor จะมีความถูกต้องและแม่นยำมากกว่า เช่น การควบคุมการเล่นเกมส์ต่างๆ โดยเฉพาะเกมส์ที่ต้องอาศัยการเคลื่อนไหวในหลาย ๆ ทิศทางที่เห็นได้ชัดเช่นบรรดาเกมส์แข่งรถทั้งหลายการที่มี Gyro Sensor มาเสริม ก็จะทำให้การควบคุมมีความถูกต้องแม่นยำมากขึ้น ไม่ว่าจะจับถือเครื่องในอิริยาบถแบบใดก็ตาม



ภาพที่ 2.1 ลักษณะของ Gyro Sensor

2.7 App Inventor

App Inventor คือ เครื่องมือที่ใช้สำหรับสร้างแอปพลิเคชันสำหรับสมาร์ทโฟนและแท็บเล็ตที่เป็นระบบปฏิบัติการ Android ซึ่งบริษัท Google ร่วมมือกับ MIT พัฒนาโปรแกรม App inventor ขึ้น ต่อมา Google ถอนตัวออกมาและยกให้ MIT พัฒนาต่อเองในนาม MIT App inventor โดย App inventor ใช้หลักการคล้ายๆ กับ Scratch แต่ซับซ้อนกว่า โดยลักษณะการเขียนโปรแกรมแบบ Visual Programming คือ เขียนโปรแกรมด้วยการต่อบล็อกคำสั่ง เน้นการออกแบบเพื่อแก้ปัญหา (problem solving) ด้วยการสร้างโปรแกรมที่ผู้เรียนสนใจบนโทรศัพท์มือถือสมาร์ทโฟน App inventor จึงเป็นอีกโปรแกรมหนึ่ง ที่เหมาะสำหรับใช้ในการสอนเขียนโปรแกรมให้นักเรียนในระดับมัธยมปลายหรือระดับมหาวิทยาลัย โดยเฉพาะผู้ที่ไม่เคยเขียนโปรแกรมมาก่อนหรือไม่ได้เรียนอยู่ในสายคอมพิวเตอร์ App Inventor servers เป็นเครื่องมือให้บริการและเก็บงานโปรเจกต์ต่างๆ ที่ผู้ใช้สร้างขึ้นมา ผู้ใช้พัฒนาโปรแกรมมือถือ Android โดยสร้างโปรเจกต์และเขียนโปรแกรมบนเว็บเบราว์เซอร์ ที่เชื่อมต่อไปยัง App Inventor servers เมื่อได้โปรแกรมมาก็สามารถทดสอบกับโปรแกรมมือถือจำลอง (Android emulator) หรือโทรศัพท์มือถือ Android

2.8 งานวิจัยที่เกี่ยวข้อง

การวิจัยของ ธวัชชัย กาญจนะทวีกุล เรื่องผลของการฝึกด้วยการกระโดดเชือกและการวิ่งระยะไกลที่มีต่อความแข็งแรงกล้ามเนื้อขาของนักมวยไทยอาชีพหญิงสังกัดค่ายมวย มหาวิทยาลัยกรุงเทพธนบุรี ผู้วิจัยมีวัตถุประสงค์เพื่อศึกษาและเปรียบเทียบผลการฝึกกระโดดเชือกและการวิ่งระยะไกลที่มีต่อความแข็งแรง กล้ามเนื้อขาของนักมวยไทยอาชีพหญิงสังกัดค่ายมวยมหาวิทยาลัยกรุงเทพธนบุรีโดย

มีกลุ่มตัวอย่างเป็นนักมวยไทยอาชีพหญิงสังกัดค่ายมวยมหาวิทยาลัยกรุงเทพธนบุรี จำนวน 30 คน ที่มีอายุระหว่าง 18-22 ปี ซึ่งมีประสบการณ์ในการฝึกซ้อมและแข่งขันมาแล้วอย่างน้อย 2 ปีที่ได้มาโดยการอาสาสมัคร จากนั้นจึงแบ่งอาสาสมัครออกเป็น 3 กลุ่มกลุ่มละ 10 คน คือ กลุ่มทดลองที่ 1 ทำการฝึกด้วยโปรแกรมการฝึกวิ่งระยะไกลเป็นเวลา 8 สัปดาห์ กลุ่มทดลองที่ 2 ทำการฝึกกระโดดเชือกเป็นเวลา 8 สัปดาห์ กลุ่มที่ 3 เป็นกลุ่มควบคุมที่ดำเนินชีวิตประจำวันตามปกติ ก่อนการฝึกทำการวัดความแข็งแรงของกล้ามเนื้อขาด้วยเครื่องวัดแรงเหวี่ยงขา (Leg Dynamometer) ทั้งสามกลุ่มเพื่อแบ่งความแข็งแรงของกล้ามเนื้อขาของอาสาสมัครในแต่ละกลุ่มให้ไม่แตกต่างกัน ($p>0.05$) จากนั้นกลุ่มทดลองที่ 1 ทำการฝึกวิ่งระยะไกล กลุ่มทดลองที่ 2 ทำการฝึกกระโดดเชือก 3 วันต่อสัปดาห์ เป็นเวลา 8 สัปดาห์ เมื่อทั้งสองกลุ่มฝึกครบ 8 สัปดาห์ จึงทำการทดสอบความแข็งแรงของกล้ามเนื้อขาด้วยเครื่องวัดแรงเหวี่ยงขา (Leg Dynamometer) หลังการฝึกอีกครั้ง ผลปรากฏว่าค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขาของกลุ่มควบคุมก่อนและหลังคือ 61.97 ± 3.69956 กก. และ 62.17 ± 2.74795 กก. ตามลำดับ ค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขาของกลุ่มทดลองที่ 1 ก่อนและหลังการฝึกคือ 61.68 ± 3.16853 กก. และ 78.51 ± 2.73839 กก. ตามลำดับค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขาของกลุ่มทดลองที่ 2 ก่อนและหลังการฝึกคือ 61.80 ± 3.14678 กก. และ 78.40 ± 3.07029 กก. ตามลำดับ เมื่อเปรียบเทียบภายในกลุ่มพบว่าค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขากลุ่มควบคุมก่อนและหลังการฝึกไม่ต่างกัน ($p>0.05$) ค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขากลุ่มทดลองที่ 1 หลังการฝึกเพิ่มขึ้น ($p<0.05$) ค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขาของกลุ่มทดลองที่ 2 หลังการฝึกเพิ่มขึ้น ($p<0.05$) เมื่อเปรียบเทียบระหว่างกลุ่มพบว่าค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขาของทั้งสามกลุ่ม มีความแตกต่างอย่างน้อยหนึ่งคู่ ($p<0.05$) เมื่อนำมาเปรียบเทียบรายคู่พบว่าความแข็งแรงกล้ามเนื้อขาของกลุ่มทดลองที่ 1 และกลุ่มทดลองที่ 2 มากกว่ากลุ่มควบคุม ($p<0.05$) ในขณะที่ค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขาของกลุ่มทดลองที่ 1 ไม่แตกต่างจากค่าเฉลี่ยความแข็งแรงกล้ามเนื้อขาของกลุ่มทดลองที่ 2 ($p>0.05$)

บทที่ 3

อุปกรณ์และวิธีดำเนินการทดลอง

ในการจัดทำโครงงานคอมพิวเตอร์ แอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ ผู้จัดทำโครงงานมีวิธีดำเนินงานโครงงานตามขั้นตอนดังต่อไปนี้

3.1 วัสดุ อุปกรณ์ และโปรแกรมที่ใช้พัฒนา

- 3.1.1 เซนเซอร์วัดความเร่งและการหมุนรอบ MPU-6050 GY-521 IMU 3-ais
- 3.1.2 ตะกั่วบัดกรีเหลว
- 3.1.3 สายไฟ
- 3.1.4 หัวแร้งบัดกรี
- 3.1.5 ถ่าน 9V และขั้วถ่าน 9V พร้อมสายไฟ
- 3.1.5 สาย USB
- 3.1.6 สายจัมเปอร์ jumper male to female ยาว 35 ซม.
- 3.1.7 โปรแกรม Arduino IDE
- 3.1.8 โปรแกรม Microsoft word
- 3.1.9 โปรแกรม Unity
- 3.1.10 โปรแกรม Visual studio
- 3.1.11 โปรแกรม Unity DevOps Version Control
- 3.1.12 โปรแกรม Canva
- 3.1.13 โปรแกรม Procreate
- 3.1.14 โปรแกรม ibisPaint
- 3.1.15 เว็บไซต์ App Inventor

3.2 ขั้นตอนการดำเนินงาน

- 3.2.1 ขั้นตอนดาวน์โหลดและเตรียมการ
 - 3.2.1.1 ดาวน์โหลดแอปพลิเคชัน unity เพื่อใช้พัฒนาเกม หลังจากนั้นสร้างบัญชีและเพิ่ม license ในแอปพลิเคชัน unity
 - 3.2.1.2 สร้างไฟล์ใน unity โดยเลือกเป็นไฟล์ 2D และตั้งชื่อเป็น jump X 2 จากนั้นเลือกที่จัดเก็บเป็นโฟลเดอร์ jumpx22
- 3.2.2 ขั้นตอนการเขียนโปรแกรม Arduino
 - 3.2.2.1 ติดตั้งโปรแกรม Arduino IDE สำหรับ Window
 - 3.2.2.2 ส่วนของ void setup ()
 - 1) เขียนคำสั่ง input ค่าจาก MPU-6050 โดยกำหนดความเร็วในการติดต่อที่ 115200
 - 2) เขียนฟังก์ชันเริ่มต้นการทำงานของเซ็นเซอร์ MPU-6050

3) ใช้คำสั่ง BLE.begin ในการเริ่มเชื่อมต่ออุปกรณ์ตรวจจับการกระโดดกับโปรแกรม Unity

3.2.2.3 ส่วนของ void loop ()

- 1) เขียนคำสั่งตรวจจับทิศทางการเคลื่อนไหวโดยใช้ข้อมูลค่ามุมจากแกน y
- 2) เขียนคำสั่งโปรแกรมการตรวจจับค่ามุมในหน่วยเรเดียนเมื่อมีมุนน้อยกว่า -1.00 ให้ส่งข้อมูลตัวเลขไปยังเกมการกระโดดอัจฉริยะ เป็นค่า “1” เมื่อมีค่ามุมจากตัวแปร y มีค่ามากกว่า 1.00 เรเดียน ให้ส่งข้อมูลตัวเลขไปยังเกมการกระโดดอัจฉริยะ Unity เป็นค่า “2” และเมื่อมีค่ามุมจากตัวแปร y มีค่าอื่น ๆ ให้ส่งข้อมูลตัวเลขไปยังเกมการกระโดดอัจฉริยะ Unity เป็นค่า “3”

3.2.3 ขั้นตอนการสร้างหน้า Menu game

3.2.3.1 Import ไฟล์ภาพหน้าเกมที่ออกแบบไว้ แล้วเปลี่ยน Texture Type เป็น Sprite (2D and UI)

3.2.3.2 เพิ่ม GameObject เป็น UI image กดเลือก Source Image เป็นไฟล์ที่ออกแบบไว้

3.2.3.3 เพิ่ม GameObject เป็น UI Button เปลี่ยนชื่อ Button เป็น play จากนั้นเปลี่ยนข้อความใน Button เป็น PLAY และตกแต่งให้สวยงาม

3.2.3.4 ทำซ้ำข้อที่ 3.2.3.2 – 3.2.3.3 โดยเปลี่ยนจาก PLAY เป็น exit

3.2.3.5 เพิ่ม GameObject เป็น Create Empty 2 อัน แล้วเปลี่ยนชื่อเป็น Scene และ Exit แล้วเขียนโปรแกรมดังภาพ 3.1-3.4 โดยคำสั่งโปรแกรมแรกตั้งชื่อคำสั่งโปรแกรมเป็น Scene และคำสั่งโปรแกรมที่สอง เป็น exit

```
1  #include <ArduinoBLE.h>
2  #include <Adafruit_MPU6050.h>
3  #include <Adafruit_Sensor.h>
4  #include <Wire.h>
5
6  float DATA;
7  const int MPU_addr = 0x68;
8
9  double temp;
10 String value;
11
12
13 Adafruit_MPU6050 mpu;
14
```

ภาพที่ 3.1 บรรทัดที่ 1-13 Arduino IDE ก่อน void setup

```
15 void setup() {
16   Serial.begin(115200);
17   Wire.begin();
18   Wire.beginTransmission(MPU_addr);
19   Wire.write(0x6B);
20   Wire.write(0);
21   Wire.endTransmission(true);
22
23
24   // done ble set up
25   //below is MPU6050 setup
26
27
28   if (!mpu.begin()) {
29     Serial.write("Failed to find MPU6050 chip");
30     while (1) {
31       delay(10);
32     }
33   }
34
35   Serial.write("MPU6050 Found!");
36
37
38   mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
39   Serial.write("Accelerometer range set to: ");
40   switch (mpu.getAccelerometerRange()) {
41     case MPU6050_RANGE_2_G:
42       Serial.write("+2G");
43       break;
44     case MPU6050_RANGE_4_G:
45       Serial.write("+4G");
46       break;
47     case MPU6050_RANGE_8_G:
48       Serial.write("+8G");
49       break;
50     case MPU6050_RANGE_16_G:
51       Serial.write("+16G");
52       break;
53   }
54   mpu.setGyroRange(MPU6050_RANGE_500_DEG);
55   Serial.write("Gyro range set to: ");
56   switch (mpu.getGyroRange()) {
57     case MPU6050_RANGE_250_DEG:
58       Serial.write("+250 deg/s");
59       break;
60     case MPU6050_RANGE_500_DEG:
61       Serial.write("+500 deg/s");
62       break;
63     case MPU6050_RANGE_1000_DEG:
64       Serial.write("+1000 deg/s");
65       break;
66     case MPU6050_RANGE_2000_DEG:
67       Serial.write("+2000 deg/s");
68       break;
69   }
70
71   mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);
72   Serial.write("Filter bandwidth set to: ");
73   switch (mpu.getFilterBandwidth()) {
74     case MPU6050_BAND_260_HZ:
75       Serial.write("260 Hz");
76       break;
77     case MPU6050_BAND_184_HZ:
78       Serial.write("184 Hz");
```

```
79         break;
80     case MPU6050_BAND_94_HZ:
81         Serial.write("94 Hz");
82         break;
83     case MPU6050_BAND_44_HZ:
84         Serial.write("44 Hz");
85         break;
86     case MPU6050_BAND_21_HZ:
87         Serial.write("21 Hz");
88         break;
89     case MPU6050_BAND_10_HZ:
90         Serial.write("10 Hz");
91         break;
92     case MPU6050_BAND_5_HZ:
93         Serial.write("5 Hz");
94         break;
95     }
96
97     Serial.write("");
98     Serial.flush();
99     delay(100);
100
101     //done MPU6050 setup
102 }
103
```

ภาพที่ 3.2 บรรทัดที่ 15-102 Arduino IDE void set up

```
104 void loop() {
105     sensors_event_t a, g, temp;
106     mpu.getEvent(&a, &g, &temp);
107     if (g.gyro.y < -1.00) {
108         Serial.write(1);
109         Serial.flush();
110         while (g.gyro.y < -1.00) {
111             Serial.write(1);
112             Serial.flush();
113             mpu.getEvent(&a, &g, &temp);
114         }
115     } else if (g.gyro.y > 1.00) {
116         Serial.write(2);
117         Serial.flush();
118     }
119     else{
120         Serial.write(3);
121         Serial.flush();
122     }
123 }
```

ภาพที่ 3.3 บรรทัดที่ 104-123 Arduino IDE void loop

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class ButtonFunction : MonoBehaviour
{
    public void restartLevel()
    {
        GameObject.Find("coin").GetComponent<count>().GameStart();
        //SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
    public void loadSceneByName(string sceneName)
    {
        SceneManager.LoadScene(sceneName);
    }
    public void ExitGame()
    {
        Application.Quit();
    }
}
```

ภาพที่ 3.4 คำสั่งโปรแกรม C# สำหรับออกเกม เปลี่ยนหน้าเกม และ รีเซ็ตารเกม

3.2.3.6 คลิก Object ชื่อ “Scene” ที่สร้างไว้ แล้วคลิก Add Component เพื่อเพิ่มโค้ดที่ชื่อ Scene ลงไป

3.2.3.7 คลิกปุ่ม play ไปที่ On Click () โดยทำการ add list เลือก object เป็น Scene แล้วเพิ่มฟังก์ชันเป็น play.LoadScene และพิมพ์ด้านล่างเป็นชื่อของหน้าที่ต้องการให้โหลด

3.2.3.8 ทำซ้ำเดิมเช่นเดียวกับปุ่ม play โดยเลือก object เป็น Exit และเพิ่มฟังก์ชันเป็น exit.ExitGame

3.2.4 ขั้นตอนการทำหน้า level select

3.2.4.1 เปลี่ยน Background ตรง Main Camera เป็นสีดำ เพิ่ม GameObject เป็น UI Button แล้วปรับขนาด หลังจากนั้นเปลี่ยนข้อความใน Button เป็น LEVEL 1 [EASY] แล้วตกแต่งให้สวยงาม

3.2.4.2 คัดลอก Button ที่สร้างไว้อีก 3 ปุ่ม แล้วเปลี่ยนข้อความใน Button เป็น LEVEL 2 [NORMAL] LEVEL 3 [HARD] และ LEVEL 4 [SURE HARD] ตามลำดับ

3.2.4.3 เพิ่ม GameObject เป็น UI Button เปลี่ยนข้อความใน Button เป็น <- BACK

3.2.4.4 เพิ่ม GameObject เป็น Create Empty ตั้งชื่อว่า levelselect Add Component ใน GameObject ที่ชื่อ levelselect เป็น คำสั่งโปรแกรม Scene

3.2.4.5 คลิกที่ปุ่ม <- BACK ไปที่ On Click () โดยทำการ Add List เลือก Object เป็น Scene Select แล้วเพิ่มฟังก์ชัน play.LoadScene พิมพ์ด้านล่างเป็นหน้าที่ต้องการให้โหลด

3.2.4.6 ทำซ้ำเดิมเหมือนข้อ 3.2.4.9 โดยเปลี่ยนเป็นปุ่ม level 1-4

3.2.4.7 เพิ่ม Gameobject เป็น UI Text เปลี่ยนข้อความใน Text เป็น select level และตกแต่ง

3.2.5 ขั้นตอนการทำหน้า level

3.2.5.1 ขั้นตอนการทำพื้นหลังเกม

1) เลือก Add New Scene แล้วสร้างโฟลเดอร์ character ในโฟลเดอร์ Assets

2) Import ไฟล์ภาพพื้นหลังเกม และภาพตัวละครทั้งหมดลงใน โฟลเดอร์ character

3) เพิ่ม GameObject เป็น UI Image เลือก Source Image เป็นภาพพื้นหลังด้าน แล้วขยายให้เท่ากับ Canvas

3.2.5.2 ทำตัว Player

1) เพิ่ม GameObject เป็น UI Image เลือก Source Image เป็นตัวละครภาพยืนนิ่ง แล้วเปลี่ยนชื่อเป็น bone

2) กดไปที่ Window แล้วเลือก Animation แล้วไปที่ Animation กด Create บันทึกชื่อไฟล์เป็น bone

3) เลือกภาพตัวละครทั้งหมดที่จะนำมาสร้างเป็นแอนิเมชัน แล้วลากลงไปในหน้าต่าง Animation จัดเรียงภาพและเวลาตามที่ต้องการ

4) กดตรง bone ในหน้าต่าง Animation แล้วเลือก create New Clip บันทึกไฟล์ชื่อว่า jump

5) กดเลือกไฟล์ภาพที่จะทำแอนิเมชันขณะตัวละครกระโดด แล้วลากลงไปในหน้าต่าง Animation จัดเรียงภาพและเวลาตามที่ต้องการ

6) กดตรงตัวละคร แล้วไปตรง Inspector ดูว่ามี Animator หรือไม่ หากไม่มีให้กด Add Component หากมีแล้ว ให้ไปตรง Controller แล้วเลือก bone

7) เลือกหน้าต่าง Animator แล้วเลือก Parameters กด + แล้วเลือก Bool ตั้งชื่อเป็น jump

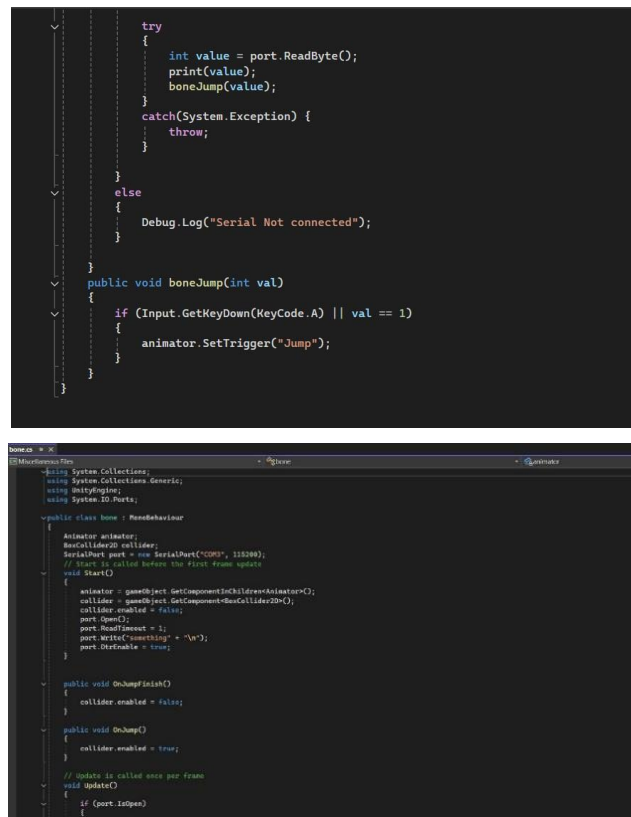
8) คลิกขวาที่ bone แล้วเลือก Make Transition ลากเส้นไปที่ jump และคลิกขวาที่ jump แล้วเลือก Make transition ลากเส้นไปที่ bone

9) กดตรงเส้นระหว่าง bone ไปหา jump กด Has Exit Time ออก ดูตรง Condition กด + จะเห็นฟังก์ชัน jump true

10) กดตรงเส้นระหว่าง jump ไปหา bone กด Has Exit Time ออก ดูตรง Condition กด + จะเห็นฟังก์ชัน jump true เปลี่ยน true เป็น false

11) สร้าง คำสั่งโปรแกรม C# ตั้งชื่อว่า bone กด Add Component เลือก Box Collider 2D แล้วกด Is Trigger

12) กด Add Component เลือก Rigidbody 2D แล้วเปลี่ยน Tag เป็น Player แล้วเขียนโปรแกรม ดังภาพ 3.5



ภาพที่ 3.5 คำสั่งโปรแกรม C# Player (ตัวละคร)

13) สร้าง Creat Empty ใน bone แล้วคำสั่งโปรแกรมลงในตัว Player

3.2.5.3 ทำเหรียญและการนับคะแนน

1) เพิ่ม UI Image กดที่ Source Image เลือกภาพเหรียญ แล้วลากเหรียญไปไว้

นอกจอ

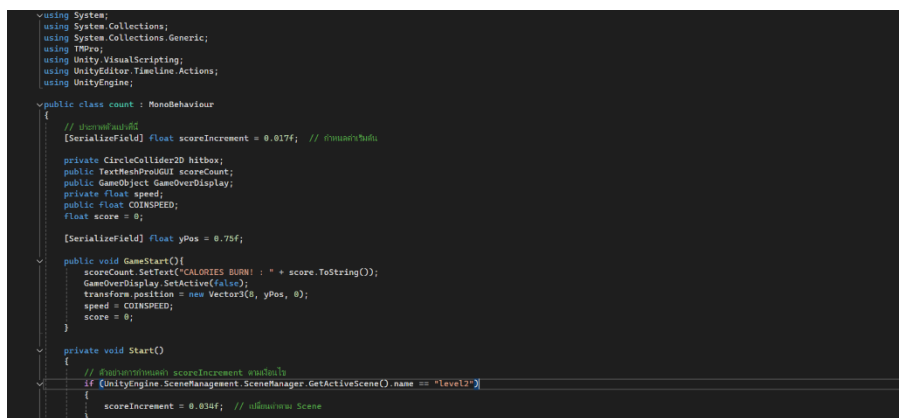
2) กด Add Component เลือก Box Collider 2D แล้วกด Is Trigger

3) สร้าง Sprite 2D เป็น Square ปรับขนาด แล้วนำไปไว้นอกจอฝั่งตรงข้ามกับเหรียญ

4) กด Add Component เลือก Box Collider 2D และ Rigidbody 2D

5) เพิ่ม UI Text - TextMeshPro แล้วเปลี่ยนเป็นคำว่า CALORIES BURN! : 0

เขียนคำสั่งโปรแกรม ดังภาพ 3.6



```
if (UnityEngine.SceneManagement.SceneManager.GetActiveScene().name == "level3")
{
    scoreIncrement = 0.085f; // เปลี่ยนตาม Scene
}

if (UnityEngine.SceneManagement.SceneManager.GetActiveScene().name == "level4")
{
    scoreIncrement = 0.17f; // เปลี่ยนตาม Scene
}

hitbox = GetComponent<CircleCollider2D>();
GameStart();

void Update()
{
    transform.position = transform.position - new Vector3(speed, 0, 0);
}

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.tag == "border")
    {
        speed = 0;
        GameOverDisplay.SetActive(true);
    }
    if (collision.gameObject.tag == "Player")
    {
        transform.position = new Vector3(8, yPos, 0);
        score += scoreIncrement; // ให้เพิ่มตาม score
        scoreCount.SetText("CALORIES BURN! : " + Math.Round(score, 3).ToString());
    }
}
```

ภาพที่ 3.6 คำสั่งโปรแกรม C# การนับเหรียญและการ Restart ค่าแคลอรี

6) เพิ่ม GameObject เป็น Create Empty ตั้งชื่อว่า game over

7) เพิ่ม UI Text – TextMeshPro 2 อัน แล้วเปลี่ยนเป็นคำว่า GAME OVER ลงใน game over

8) เพิ่ม UI Button แล้วเปลี่ยน Text เป็นคำว่า RESTART ลงใน game over กด + ที่ On Click () เปลี่ยน Object เป็น restart เปลี่ยน function เป็น ButtonFunction RestartLevel()

9) เพิ่ม UI Button แล้วเปลี่ยน Text เป็นคำว่า MENU ลงใน game over กด + ที่ On Click () เปลี่ยน Object เป็น restart เปลี่ยน function เป็น ButtonFunction loadScreenByName() เปลี่ยนชื่อ เป็น Menu แล้วเพิ่ม คำสั่งโปรแกรม C# เหมือนในภาพที่ 3.4

10) เพิ่ม Scripts ลงในเหรียญ แล้วเปลี่ยน COINSPEED เป็น 0.003

11) กดคัดลอก GameObject ทั้งหมด

12) กด Add New Scene แล้ววาง GameObject ลงใน Scene

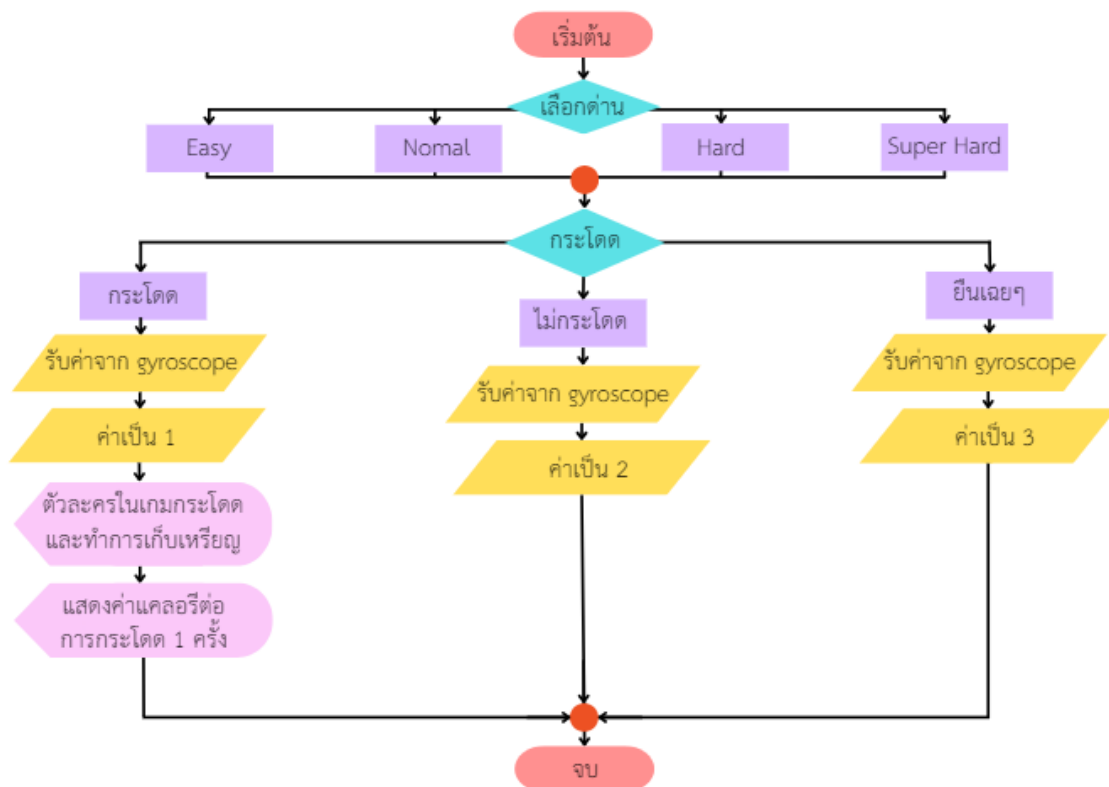
13) ทำซ้ำข้อ 14-16 ให้ครบ 4 ด้าน โดยเปลี่ยน COINSPEED เป็น 0.005, 0.015 และ 0.045 ตามลำดับ

3.2.6 ขั้นตอนการประดิษฐ์ตัวอุปกรณ์ตรวจจับการกระโดด

3.2.6.1 ออกแบบอุปกรณ์ให้มีความเหมาะสมต่อเกม โดยใช้อุปกรณ์เป็นสนับเข่า

3.2.6.2 ตัดเอาฟองน้ำที่อยู่ในสนับเข่าตรงบริเวณเข่าออก โดยตัดสนับเข่าออกแค่บริเวณด้านบน แล้วเย็บแถบหนามเตยลงไปสนับเข่า แล้วนำบอร์ดใส่เข้าไปในช่องว่างแทนฟองน้ำ

3.2.6.3 เย็บถุงให้กับถ่าน 9V โดยเย็บแถบหนามเตยเข้าไปในบริเวณด้านบนของถุง หลังจากนั้นใส่ถ่านลงไป และเย็บถุงติดกับสนับเข่า



บทที่ 4

ผลการทดลอง

การสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ มีวัตถุประสงค์เพื่อสร้างแอปพลิเคชันและสร้างเกมกระโดดเชือกอัจฉริยะจากการเขียนโปรแกรมผ่านเซนเซอร์ตรวจจับและวัดค่ามุม MPU-6050 เพื่อทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดดและเพื่อสำรวจความพึงพอใจของผู้ใช้งานเกมกระโดดเชือกอัจฉริยะ ผู้ทำโครงการได้สรุปและอภิปรายผล ดังนี้

4.1 การสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ

จากการศึกษาการวัดค่าโดยใช้เซนเซอร์ตรวจจับและวัดค่ามุม จากนั้นเขียนคำสั่งโปรแกรมตรวจจับค่ามุมบน Arduino IDE เพื่อวัดค่ามุมเป็นหน่วยเรเดียน แล้วแปลงค่ามุมเป็นตัวเลข หลังจากนั้นเขียนคำสั่งโปรแกรมภาษา C# ในเกม โดยใช้ Visual Studio ส่งค่าที่แปลงแล้วจากบอร์ด Arduino ให้กับเกมเพื่อใช้เป็นคำสั่งการเคลื่อนไหวภายในเกม สามารถนำไปสร้างเกมกระโดดเชือกอัจฉริยะได้

จากการศึกษาค่า Basal Metabolic Rate และ Total Daily Energy Expenditure ทำให้สามารถนำมาคำนวณหาพลังงานที่ควรเผาผลาญและพลังงานที่แนะนำต่อวัน แล้วนำไปประยุกต์สร้างแอปพลิเคชันคำนวณแคลอรีจาก App Inventor ได้



ภาพที่ 4.1 การทดสอบเกมกระโดดเชือกอัจฉริยะ

4.2 การทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดด

จากการทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดดจำนวน 100 ครั้ง พบว่าอุปกรณ์สามารถตรวจจับการกระโดดของผู้ใช้งานได้ถูกต้องจำนวน 90 ครั้ง จาก 100 ครั้ง คิดเป็นความแม่นยำร้อยละ 90.00 มีความแม่นยำอยู่ในระดับมากที่สุด โดยมีเกณฑ์การแปลความหมายร้อยละ ได้แก่

80.00 - 100.00	หมายถึง แม่นยำมากที่สุด
60.00 - 79.99	หมายถึง แม่นยำมาก
40.00 - 59.99	หมายถึง แม่นยำปานกลาง
20.00 - 39.99	หมายถึง แม่นยำน้อย
00.00 - 19.99	หมายถึง แม่นยำน้อยที่สุด

4.3 การประเมินความพึงพอใจของผู้ใช้งานแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ

ตารางที่ 4.1 ผลการสำรวจความพึงพอใจที่มีต่อโครงการจากกลุ่มผู้ใช้งาน

หัวข้อในการประเมิน	ระดับความพึงพอใจจากผู้ใช้งาน 100 คน		
	ค่าเฉลี่ย (x)	แปลผล	ส่วนเบี่ยงเบนมาตรฐาน (SD)
1. ความสนุกสนานในการเล่นเกมกระโดดเชือกอัจฉริยะ	4.64	มากที่สุด	0.594928
2. ประสิทธิภาพในการตรวจจับของเซนเซอร์	4.80	มากที่สุด	0.492366
3. ความสวยงามของหน้าตาเกม	4.51	มากที่สุด	0.731610
4. ความหลากหลายของเกมกระโดดเชือก	4.41	มาก	0.809420
5. ความพึงพอใจโดยรวมที่มีต่อเกมกระโดดเชือกอัจฉริยะ	4.66	มากที่สุด	0.623123

เกณฑ์การแปลความหมายค่าเฉลี่ย

4.50 - 5.00	หมายถึง มีความพึงพอใจมากที่สุด
3.50 - 4.49	หมายถึง มีความพึงพอใจมาก
2.50 - 3.49	หมายถึง มีความพึงพอใจปานกลาง
1.50 - 2.49	หมายถึง มีความพึงพอใจน้อย
1.00 - 1.49	หมายถึง มีความพึงพอใจน้อยที่สุด

บทที่ 5

สรุปและอภิปรายผลการทดลอง

การสร้างแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ มีวัตถุประสงค์เพื่อสร้างแอปพลิเคชันและสร้างเกมกระโดดเชือกอัจฉริยะจากการเขียนโปรแกรมผ่านเซนเซอร์ตรวจจับและวัดค่ามุม MPU-6050 เพื่อทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดดและเพื่อสำรวจความพึงพอใจของผู้ใช้งานเกมกระโดดเชือกอัจฉริยะ ได้ผลการทดลอง ดังนี้

5.1 สรุปผลการทดลอง

5.1.1 จากการศึกษาการวัดค่าจากเซนเซอร์ตรวจจับและวัดค่ามุม สามารถเขียนคำสั่งโปรแกรมและแปลงค่าจากมุมเรเดียนเป็นตัวเลข เพื่อสร้างเกมกระโดดเชือกอัจฉริยะได้ และจากการศึกษาค่า Basal Metabolic Rate และ Total Daily Energy Expenditure ทำให้สามารถคำนวณหาพลังงานที่ควรเผาผลาญและพลังงานที่แนะนำต่อวัน เพื่อสร้างแอปพลิเคชันคำนวณแคลอรีได้

5.1.2 จากการทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดด พบว่าอุปกรณ์มีความแม่นยำคิดเป็นร้อยละ 90.00 ซึ่งถือว่ามีความแม่นยำในระดับมากที่สุด

5.1.3 จากการสำรวจความพึงพอใจของผู้ใช้งานเกมกระโดดเชือกอัจฉริยะ จำนวน 100 คน พบว่า ผู้ใช้งานมีความพึงพอใจด้านความสนุกสนาน ประสิทธิภาพในการตรวจจับของเซนเซอร์ ความสวยงามของหน้าตาเกม และความพึงพอใจโดยรวมที่มีต่อเกมกระโดดเชือกอัจฉริยะอยู่ในระดับพึงพอใจมากที่สุด ส่วนความหลากหลายของเกมอยู่ในระดับมาก

5.2 อภิปรายผลการทดลอง

5.1.1 จากการศึกษาการวัดค่าโดยใช้เซนเซอร์ตรวจจับและวัดค่ามุม และเขียนคำสั่งโปรแกรมการตรวจจับค่ามุมในหน่วยเรเดียนเพื่อวัดค่ามุมการงอเข้า หลังจากนั้นเขียนโปรแกรม ภาษา C# เพื่อส่งค่าที่แปลงแล้วให้กับเกมเพื่อใช้เป็นคำสั่งการเคลื่อนไหว สามารถนำมาประยุกต์เกมกระโดดเชือกอัจฉริยะได้ และจากการรับค่าข้อมูลทั่วไปของผู้ใช้งาน ได้แก่ เพศ น้ำหนัก ส่วนสูง และอายุ สามารถนำมาใช้คำนวณเพื่อหาค่า BMR ได้ และจากการรับค่าข้อมูลประเภทกิจกรรมของผู้ใช้งานแต่ละคนสามารถนำมาใช้คำนวณต่อเพื่อหาค่า TDEE จากนั้นสามารถนำค่า BMR และ TDEE มาคำนวณหาพลังงานที่ควรเผาผลาญและพลังงานที่แนะนำต่อวัน เพื่อประยุกต์สร้างแอปพลิเคชันคำนวณแคลอรีจาก App Inventor ได้

5.1.2 จากการทดสอบความแม่นยำในการรับค่าของอุปกรณ์ตรวจจับการกระโดด พบว่าเมื่อผู้ใช้งานมีการเคลื่อนไหว ได้แก่ กระโดด หยุดกระโดด และยืนนิ่ง เซนเซอร์ตรวจจับและวัดค่ามุมภายในอุปกรณ์สามารถรับค่าเป็น 1, 2 และ 3 ตามลำดับ แล้วส่งค่าแบบไร้สายไปที่โปรแกรม Unity ทำให้เกมแสดงผลได้ถูกต้องและนำไปคำนวณต่อได้ ซึ่งส่งผลให้อุปกรณ์มีความแม่นยำในระดับมากที่สุด สามารถนำไปใช้ในการออกกำลังกายในชีวิตประจำวันได้

5.1.3 จากการสำรวจความพึงพอใจของผู้ใช้งานเกมกระโดดเชือกอัจฉริยะ จำนวน 100 คน พบว่าผู้ใช้งานส่วนใหญ่รู้สึกสนุกสนานไปกับการเล่นเกม พึงพอใจในประสิทธิภาพในการตรวจจับของ

เซนเซอร์ในอุปกรณ์ตรวจจับการกระโดด ประทับใจในความสวยงามของหน้าต่างเกมและความหลากหลายของเกม โดยค่าเฉลี่ยความพึงพอใจโดยรวมต่อเกมกระโดดเชือกอัจฉริยะอยู่ที่ 4.66 หรือพึงพอใจมากที่สุด และมีส่วนเบี่ยงเบนมาตรฐาน 0.623123

5.3 ข้อเสนอแนะ

5.3.1 สร้างเกมให้มีฟังก์ชันมากขึ้น เช่น การจับค่าอัตราการเต้นของหัวใจ การสร้างบัญชีในเกม และการสรุปผลการเล่นเกมในแต่ละวันและบันทึกค่าแคลอรี

5.3.2 เพิ่มตัวแปรในการนับแคลอรี เช่น ความต่อเนื่อง และน้ำหนักของผู้เล่น

5.3.3 สร้างแอปพลิเคชันให้สามารถใช้ในระบบปฏิบัติการ IOS ได้

บรรณานุกรม

- ธวัชชัย กาญจนะทวิกุล. (2563). ผลของการฝึกด้วยการกระโดดเชือกและการวิ่งระยะไกลที่มีต่อความแข็งแรงกล้ามเนื้อขาของนักมวยไทยอาชีพหญิง. สืบค้น 4 ธันวาคม 2566, จาก https://research.bkkthon.ac.th/abstac/ab_02092563141724.pdf
- โรงพยาบาลนวเวช. (ม.ป.ป.). ประโยชน์ของการออกกำลังกาย. สืบค้น 4 ธันวาคม 2566, จาก <https://navavej.com/articles/18788#>
- ลงทุนเกิร์ล. (2564). Procreate แอปวาดภาพ อันดับ 1 ใน iPad ที่ปฏิเสธคำเชิญไป Silicon Valley. สืบค้น 4 มิถุนายน 2567, จาก <https://www.longtungirl.com/3167>
- วสันต์ทองดี. (2562). App Inventor คืออะไร?. สืบค้น 4 ธันวาคม 2566, จาก <https://programmingappinventor.wordpress.com/>
- Cleveland Clinic. (2567). Hop to It: 6 Benefits of Jumping Rope. สืบค้น 26 มกราคม 2567, จาก <https://health.clevelandclinic.org/jump-rope-benefits>
- Criclabs Digital Solution Agency in Thailand. (ม.ป.ป.). วิธีเริ่มต้นพัฒนา Android App จากการเขียน Mobile App ด้วย Visual Studio. สืบค้น 4 ธันวาคม 2566, จาก <https://www.criclabs.co/post/mobile-app-by-visual-studio>
- Fit Me Sportswear. (ม.ป.ป.). เคล็ดลับวิธี คำนวณปริมาณแคลอรี ที่ต้องการต่อวัน. สืบค้น 4 ธันวาคม 2566, จาก <https://www.fitmesport.com/calculate-calories-per-day/>
- Hfocus. (2566). ปี 2566 กรมควบคุมโรค สรุป 3 อันดับแรกโรคระบาดมากที่สุด. สืบค้น 31 ธันวาคม 2566, จาก <https://www.hfocus.org/content/2023/12/29387>
- KongRuksiam Studio. (2567). รู้จักกับ Unity Game Engine สำหรับการพัฒนาเกม 2 มิติ / 3 มิติ. สืบค้น 1 มีนาคม 2567, จาก <https://kongruksiam.medium.com/>
- Thaimobilecenter. (2564). รู้จักเหล่าเซ็นเซอร์อัจฉริยะบนสมาร์ทโฟน มีอะไรบ้าง แต่ละชนิดทำงานอย่างไร ?. สืบค้น 4 ธันวาคม 2566, จาก <https://www.thaimobilecenter.com/article-2557/understanding-sensors-on-mobile-phone-and-smartphone.asp>
- The Invention. (2564). Arduino IDE คืออะไร มีวิธีการติดตั้งโปรแกรมอย่างไร และการใช้โปรแกรมยังงัยกันนะ. สืบค้น 4 ธันวาคม 2566, จาก <https://www.ai-orporation.net/2021/11/18/what-is-arduino-ide/>
- World Health Organization. (2566). Physical activity. สืบค้น 4 ธันวาคม 2566, จาก <https://www.who.int/news-room/fact-sheets/detail/physical-activity>

ภาคผนวก

ภาคผนวก ก
อุปกรณ์ตรวจจับการกระโดด



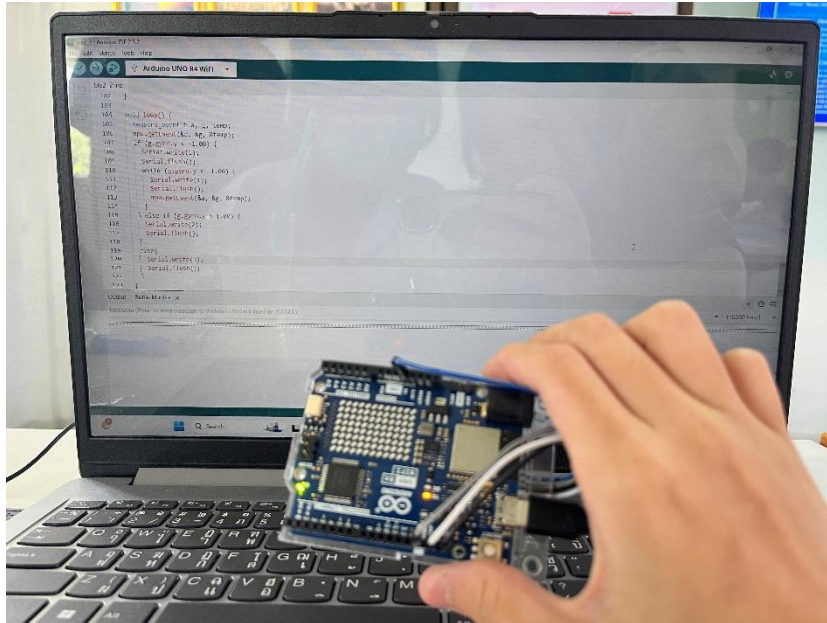
ภาพที่ 1 อุปกรณ์ตรวจจับการกระโดด



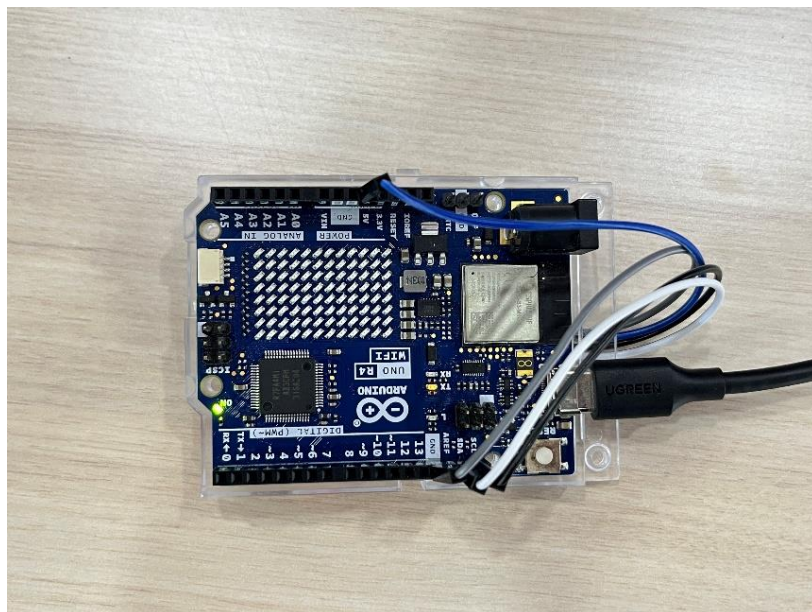
ภาพที่ 2 การทดสอบอุปกรณ์ตรวจจับการกระโดด

ภาคผนวก ข

ขั้นตอนการเขียนโปรแกรม Arduino IDE



ภาพที่ 3 การทดสอบการอ่านค่าจากเซ็นเซอร์ตรวจจับและวัดค่ามุม MPU-6050



ภาพที่ 4 การต่อวงจร Gyro Sensor MPU-6050 กับบอร์ด Arduino

ภาคผนวก ค
หน้าต่างแอปพลิเคชันกระโดดเชือกอัจฉริยะ



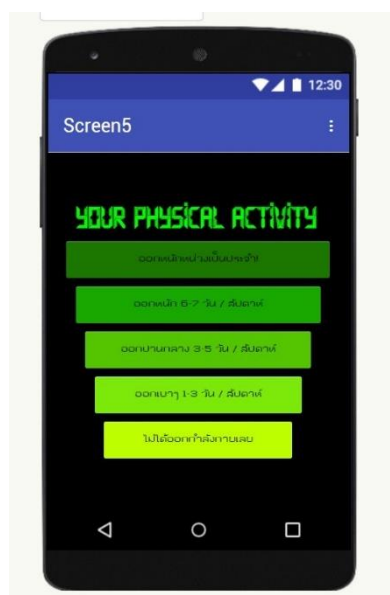
ภาพที่ 5 หน้าต่าง Menu



ภาพที่ 6 หน้าต่างเลือกเพศ



ภาพที่ 7 หน้าต่างคำนวณค่า BMR



ภาพที่ 8 หน้าต่างคำนวณค่า TDEE



ภาพที่ 9 หน้าต่างคำนวณพลังงานที่ต้องเผาผลาญและพลังงานที่แนะนำต่อวัน



ภาพที่ 10 หน้าต่างแสดงผลพลังงานที่ต้องเผาผลาญและพลังงานที่แนะนำต่อวัน

ภาคผนวก ง
หน้าตาเกมกระโดดเชือกอัจฉริยะ



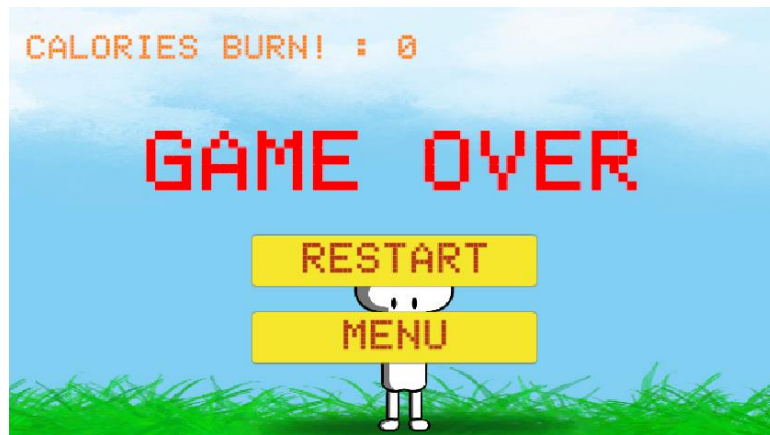
ภาพที่ 11 หน้าต่าง menu



ภาพที่ 12 หน้าต่างเลือกระดับความยาก



ภาพที่ 13 หน้าต่างภายในเกม



ภาพที่ 14 หน้าต่าง game over

ภาคผนวก จ

ผลการประเมินความพึงพอใจของผู้ใช้งานแอปพลิเคชันและเกมกระโดดเชือกอัจฉริยะ

ตารางที่ 1 ผลการสำรวจร้อยละความพึงพอใจที่มีต่อโครงการจากกลุ่มผู้ใช้งาน

หัวข้อในการประเมิน	ระดับความพึงพอใจจากผู้ใช้งาน 100 คน (%)				
	มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด
1. ความสนุกสนานในการเล่นเกมกระโดดอัจฉริยะ	70	24	6	-	-
2. ประสิทธิภาพในการตรวจจับของเซนเซอร์	84	12	4	-	-
3. ความสวยงามของหน้าตาเกม	64	24	11	1	-
4. ความหลากหลายของเกมกระโดดเชือกอัจฉริยะ	56	32	9	3	-
5. ความพึงพอใจโดยรวมที่มีต่อเกมกระโดดเชือก	74	18	8	-	-