เกมปาลูกบอล

"ON THE BALL" GAMES PLAYER

รักษิณา ยมาสิริ

วรเดช กนกวรรณรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เกมปาลูกบอล

"ON THE BALL" GAMES PLAYER

T144367

รักษิณา ษมาสิริ

วรเดช กนกวรรณรัตน์

สจหมู่.....
เลขทะเบียน 144367
วัน.เดือน.ปี 24 พ.ย. 2559

.b 12817444X
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เกมปาลูกบอล

" ON THE BALL " GAMES PLAYER

ผู้จัดทำ

1. นางสาวรักษิณา ษมาสิริ รหัสนักศึกษา 55011025

2. นายวรเดช กนกวรรณรัตน์ รหัสนักศึกษา 55011074

..... อาจารย์ที่ปรึกษา

(ดร. ธนัญชัย ตรีภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกมปาลูกบอล

นางสาวรักษิณา ษมาสิริ 55011025
นายวรเดช กนกวรรณรัตน์ 55011074
ดร.ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษา
ปีการศึกษา 2558

บทคัดย่อ

จากยุคสมัยที่เปลี่ยนไป ทำให้กิจกรรมยามว่างของเด็กไทยส่วนใหญ่ กลายเป็นการเล่นเกมผ่านคอมพิวเตอร์หรืออุปกรณ์ต่างๆ ที่ใช้เพียงการขยับนิ้วมือเพื่อเล่นเกมในท่าทางเดิมๆ แม้จะเป็นการพัฒนาสมองของเด็ก แต่การอยู่ในท่าทางเดิมๆ เป็นเวลานาน จะก่อให้เกิดผลเสียทางด้านสุขภาพและพัฒนาการ อาจเกิดอาการสายตาพร่ามัว ปวดศีรษะ ทั้งยังทำให้โอกาสในการออกกำลังกายลดลง

ทางคณะผู้จัดทำจึงเห็นควรจะนำเกมที่เด็กๆ ชอบ มาพัฒนาให้เป็นเกมที่ก่อให้เกิดพัฒนาการของร่างกาย ทั้งทางด้านกล้ามเนื้อ สมอง การใช้ความคิดในการแก้ปัญหา จึงเกิดเกมนี้ขึ้น โดยจะฉายภาพเกมที่มีลักษณะเป็นเกมยิงบนผนัง และให้ผู้เล่นทำการเล่นโดยโยนลูกบอลกระทบผนังแทนการยิง ภายใต้ชื่อโครงงาน “เกมปาลูกบอล” ถือว่าเป็นการออกกำลังกายไปในตัว ไม่ใช่การเล่นเกมด้วยอิริยาบถเดิมๆเป็นเวลานานอีกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

" ON THE BALL " GAMES PLAYER

Ms. Raksina Samasiri 55011025

Mr. Woradej Kanokwannarat 55011074

Dr. Thanunchai Threepak Advisor

Academic Year 2015

ABSTRACT

From the times changed. Thai children's activities also changed to play video games in computer or digital devices. They use only their fingers doing the same motion over and over in the same manner to play video games. Though playing video games develops child's brain, but it can cause adverse effects on their health and their growth, such as blurred vision, headache and it decreases the frequency, time of their exercise.

Then, we thought it's appropriate to create a better video game player, " On the ball " games player, improving both physical and brain development in children since this game, played by throwing the ball to the screen, makes player to work their abs while playing it. " On the ball " games player is the new way to improve the child development so parents don't have to worry about child physical development problem anymore.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงไปได้ หากมิได้รับความกรุณาจากอาจารย์ ด็อกเตอร์ ธนัญชัย ตรีภาค อาจารย์ที่ปรึกษาโครงงาน ซึ่งเป็นผู้ให้การสนับสนุนในด้านอุปกรณ์และให้คำปรึกษาเกี่ยวกับการทำงานเป็นอย่างดี ตลอดจนให้คำแนะนำในการแก้ไขข้อบกพร่องต่างๆมาโดยตลอด คณะผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างมากในความอนุเคราะห์จากอาจารย์ที่คอยให้การสนับสนุนในการทำปริญญานิพนธ์นี้เสมอมา อีกทั้งอาจารย์อัครเดช วัชระภูพงษ์ ที่ช่วยเหลือ ให้ความรู้ในการทำงานให้สำเร็จได้ด้วยดี รวมถึงห้องวิจัย ISAG (Information Security Advisory Group) ของภาควิชาวิศวกรรมคอมพิวเตอร์ ที่เอื้อเฟื้อสถานที่ ให้เป็นสถานที่ทำงาน ศึกษาหาความรู้ประกอบการทำโครงงานในครั้งนี้ คุณพ่อคุณแม่ ครอบครัว และเพื่อนๆ ที่คอยให้คำปรึกษาสนับสนุนในเรื่องต่างๆ เป็นกำลังใจที่ยิ่งใหญ่มาโดยตลอด ทำให้คณะผู้จัดทำคลายเครียดจากการทำงานได้อยู่เสมอ

คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง ในความกรุณาของทุกท่านที่ได้กล่าวมานี้ และหวังเป็นอย่างยิ่งว่า ปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อทุกท่านที่ต้องการศึกษาทางด้านนี้ และสามารถเป็นคำแนะนำให้แก่นักศึกษารุ่นต่อๆไปได้

รักษิณา ษมาสิริ

วรเดช กนกวรรณรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 1

1.4 วิธีการดำเนินการ 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

1.6 ส่วนประกอบของปริญญานิพนธ์ 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 ภาษา Python 4

2.2 Image Processing 4

2.3 Library OpenCV 5

2.4 ระบบสี RGB และ HSV 7

2.5 Edge Detection 9

2.6 Multi-Threads 13

2.7 Pyglet Library 14

บทที่ 3 การออกแบบและพัฒนา 15

3.1 รายละเอียดของระบบ 15

3.2 เครื่องมือที่ใช้พัฒนา 16

3.3 โครงสร้างของระบบ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.4 การออกแบบและผังการทำงานของระบบ 18
3.5 ขั้นตอนการทำงานของระบบ 23
3.6 การออกแบบ Interface ของเกม 24
3.7 การพัฒนาส่วนตรวจจับ 25
3.8 การพัฒนาตัวเกมและ Use Case Diagram 26
3.9 การพัฒนาการทำงานร่วมกันของส่วนตรวจจับและตัวเกม 29

บทที่ 4 การทดลองและผลการทดลอง 30
4.1 การทดลองการตรวจจับลูกบอล 30
4.2 การทดลองตรวจจับการกระทบผนังของลูกบอล 36
4.3 การทดลองปรับปรุงการตรวจจับการกระทบผนังโดยใช้ทิศทางการเคลื่อนที่ของลูกบอล 39
4.4 การทดลองตรวจจับ Laser Pointer 40
4.5 การทดลองแบ่ง Thread 42
4.6 การทดลองพัฒนาเกมด้วย Pyglet Library 44

บทที่ 5 บทสรุปและข้อเสนอแนะ 53
5.1 สรุปและบทวิจารณ์ 53
5.2 ปัญหาและอุปสรรค 53
5.3 แนวทางแก้ไขปัญหาและอุปสรรค 53
5.4 แนวทางการพัฒนาต่อ 53

บรรณานุกรม 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง หน้า

4.1 ค่าสี HSV ของบอลแต่ละสี.....35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป หน้า

2.1 การทำงานของ Image Processing 4
2.2 ระบบสี RGB 7
2.3 ระบบสี HSV 8
3.1 โครงสร้างของอุปกรณ์ในระบบ 15
3.2 System Architecture 17
3.3 แผนภาพ BPMN ของหน้าเลือกเกม 20
3.4 แผนภาพ BPMN ของเกมเบื้องต้นเกมที่ 1 " เกมพื้นฐาน " 22
3.5 Flowchart ของการตรวจจับลูกบอล 23
3.6 หน้าเลือกเกม 24
3.7 หน้าเลือกเกมเมื่อเลือกเกมต่างๆ จากแถบเกมเบื้องต้น 24
3.8 การตั้งค่าภายในเกม 25
3.9 Use Case Diagram ของเกมพื้นฐาน 26
3.10 Use Case Diagram ของเกมปาเป้า 27
3.11 Use Case Diagram ของเกมปาแมลงวัน 28
4.1 Track Bar สำหรับปรับค่าต่างๆ ของ Hough Transform ในฟังก์ชั่น 30
4.2 ผลลัพธ์การตรวจจับ 31
4.3 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น 31
4.4 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold มากขึ้นอีก 32
4.5 ใช้ GaussianBlur เพื่อลด Noise ที่เกิดขึ้นในภาพที่นำมาตรวจจับ 32
4.6 ตรวจเจอเส้นโค้งที่ไม่ใช่วงกลม 33
4.7 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น 33
4.8 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น 34
4.9 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น 35
4.10 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น 35
4.11 เมื่อนำพื้นที่ที่ได้มาเข้าฟังก์ชั่น HoughCircles 36
4.12 ลูกบอลถูกตรวจจับขณะที่อยู่ติดผนัง 37
4.13 รูปการ์ตูนขึ้นมาบนจอภาพ หลังจากลูกบอลกระทบผนัง 38
4.14 การนับจำนวนบอลที่เข้ามาใหม่ใน Console 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป หน้า

4.15 การตรวจจับการเคลื่อนที่ของลูกบอลในทิศทางต่างๆ 39
4.16 รูปการตรวจจับ Laser Pointer 41
4.17 รูปผลลัพธ์ของการปรับขนาดการตรวจจับ Laser Pointer 41
4.18 ส่วนตรวจจับที่เบลอเมื่อโยนบอล 42
4.19 ส่วนตรวจจับที่ทำงานได้เร็วขึ้น 43
4.20 เกมที่ทำงานได้ 44
4.21 หน้าจอเกมที่ได้จากโปรแกรม 4.1 45
4.22 หน้าจอเกม " เกมพื้นฐาน " เบื้องต้น 46
4.23 หน้าจอเกม " เกมพื้นฐาน " หลังจากลูกบอลกระทบหน้าจอ 46
4.24 หน้าจอเกม " เกมปาเป้า " เบื้องต้น 47
4.25 หน้าจอเกม " เกมปาเป้า " เมื่อปรับขนาดของเป้า 48
4.26 หน้าจอเกม " เกมปาเป้า " เมื่อปรับแถบ Hit(s) / Score 48
4.27 หน้าจอเกม " เกมปาเป้า " เมื่อปาลูกบอลโดนเป้า 49
4.28 หน้าจอก่อนเริ่ม " เกมปาแมลงวัน " 49
4.29 หน้าจอเกม " เกมปาแมลงวัน " เบื้องต้น 50
4.30 หน้าจอเกม " เกมปาแมลงวัน " เมื่อปรับแถบจำนวนแมลงวัน 50
4.31 หน้าจอเกม " เกมปาแมลงวัน " เมื่อเปิดและปรับแถบตัวนับเวลา 51
4.32 หน้าจอเกม " เกมปาแมลงวัน " เมื่อโยนลูกบอลโดนแมลงวัน 51
4.33 หน้าจอเกม " เกมปาแมลงวัน " เมื่อกดปุ่ม Pause 52
4.34 หน้าจอเกม " เกมปาแมลงวัน " เมื่อหมดเวลา 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ยุคสมัยที่เปลี่ยนไป ทำให้กิจกรรมยามว่างของเด็กไทยส่วนใหญ่ กลายเป็นการเล่นเกมผ่านคอมพิวเตอร์ หรืออุปกรณ์ต่างๆ ที่ใช้เพียงการขยับนิ้วมือเพื่อเล่นเกม ภายใต้ในท่าทางเดิมๆ การเล่นเกมคอมพิวเตอร์เช่นนี้ แม้จะเป็นการพัฒนาสมองของเด็ก แต่การอยู่ในท่าทางเดิมๆ เป็นเวลานานจะก่อให้เกิดผลเสียทางด้านสุขภาพ อาจเกิดอาการสายตาพร่ามัว ปวดศีรษะ ปวดหลัง ข้ออักเสบ จากการนั่งจ้องมองจอภาพดิจิทัลในอิริยาบทเดิมเป็นเวลานานๆ และเกิดผลเสียทางด้านพัฒนาการของร่างกาย ทั้งยังทำให้โอกาสในการออกกำลังกายลดลงอีกด้วย

ทางคณะผู้จัดทำเล็งเห็นประโยชน์ของการเล่นเกม ภายใต้ข้อเสียในการควบคุมเกม ที่ก่อให้เกิดปัญหาสุขภาพ คณะผู้จัดทำจึงเห็นควรจะนำเกมที่เด็กๆ ชอบ มาพัฒนาเป็นเกมที่ก่อให้เกิดพัฒนาการทั้งทางด้านสุขภาพร่างกาย สมอง รวมถึงการใช้ความคิดในการแก้ปัญหา จึงเกิดโครงงาน “เกมปาลูกบอล” นี้ขึ้น โดยเป็นเกมที่ใช้การปาลูกบอลในการควบคุมการเล่นเกม ถือว่าเป็นการออกกำลังกายไปในตัว ไม่ใช่แค่การเล่นเกมในอิริยาบถเดิมๆ อีกต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาเกมที่ผู้เล่นสามารถเล่นเกมไปพร้อมกับเคลื่อนไหวร่างกาย ลดปัญหาสุขภาพจากการเล่นเกมคอมพิวเตอร์ทั่วไป และสร้างเสริมสุขภาพที่ดีแก่ผู้เล่น
2) เพื่อศึกษาการประมวลผลภาพ และนำมาประยุกต์ใช้ในการสร้างโครงงาน
3) เพื่อศึกษาและฝึกทำงานด้านเอกสาร คือการทำ Requirement analysis และ Design document ของการสร้างชิ้นงาน
4) เพื่อเป็นต้นแบบเครื่องเล่นเกม ให้ผู้อื่นที่สนใจสร้างเกมมาใช้ร่วมกับเครื่องเล่นนี้ต่อไป

1.3 ขอบเขตของโครงงาน

1) อุปกรณ์ต่างๆ สามารถทำงานร่วมกันเป็นระบบควบคุมเกมได้
2) มีระบบควบคุมเกม และเกมต่างๆ ทำงานได้ตามรายละเอียดของระบบ
3) ชุดติดตั้งโปรแกรมที่สามารถติดตั้งเกมดังกล่าวในเครื่องคอมพิวเตอร์อื่นๆ ได้
4) มีรายการเกมเบื้องต้น 3 รายการ ได้แก่
    a) เกมพื้นฐาน เพื่อทดสอบระบบ เมื่อโยนลูกบอลกระทบผนังจะเกิดรูปบนผนัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b) เกมปาเป้า จะมีเป้าบนผนัง เมื่อโยนลูกบอลกระทบผนังตามจำนวนครั้งที่กำหนด จะนับคะแนนให้ผู้เล่น

c) เกมปาแมลงวัน จะมีแมลงวันบินอยู่บนผนัง เมื่อโยนลูกบอลโดนแมลงวันจะนับคะแนน

1.4 วิธีการดำเนินการ

1) วางแผนการทำงาน โดยแบ่งงานออกเป็นส่วนย่อยๆ
2) ติดตั้งโปรแกรม Bizagi และศึกษาวิธีการใช้งาน
3) ออกแบบ Interface และ Flow ของหน้าเลือกเกม
4) เขียนแผนภาพ BPMN ที่แสดงขั้นตอนการทำงานของระบบ ด้วยโปรแกรม Bizagi
5) ติดตั้งกล้อง Webcam Logitech c920 เพื่อทดลองใช้งาน
6) ลงโปรแกรม Pycharm เพื่อใช้เขียนหน้าเลือกเกมด้วยภาษา Python
7) ศึกษาการเขียนซอร์สโค้ดด้วยภาษา Python
8) ศึกษา Library OpenCV
9) ทดลองเขียนโปรแกรมตรวจจับขอบของวัตถุที่เป็นวงกลม
10) หาลูกบอลขนาดเล็ก ที่มีสีสันต่างๆ มาทดสอบเขียนโปรแกรมตรวจจับ
11) เลือกสีลูกบอลที่จะนำมาใช้ในการเล่นเกม ได้แก่ สีส้ม สีน้ำเงิน สีเขียว และสีเหลือง
12) ทำให้การตรวจจับลูกบอลแม่นยำขึ้น โดยการกำหนดสีของลูกบอลที่ใช้
13) ทดลองตรวจจับตำแหน่งที่ลูกบอลกระทบผนัง
14) เขียนตัวทดลองของเกมพื้นฐาน เพื่อทดสอบการตรวจจับบอลกระทบกับผนังของระบบ
15) สรุปผลการทดลองและจัดทำเอกสาร

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) เด็กๆ มีสุขภาพร่างกายที่ดีขึ้น จากการเล่นเกมที่ได้ออกกำลังกาย โดยเคลื่อนไหวร่างกายไปด้วยในขณะเล่น
2) ได้พัฒนาทักษะทางด้านการประมวลผลภาพ และการนำมาประยุกต์ใช้
3) ได้ฝึกฝนการเขียนโปรแกรมด้วยภาษา Python
4) ได้ความรู้จากการทำ Requirement Analysis และ Design Document ของโครงงาน

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน ประกอบด้วยภาษา Python, Image Processing, Library OpenCV, ระบบสี RGB และ HSV

บทที่ 3 การออกแบบและพัฒนา กล่าวถึงรายละเอียดของโครงงานนี้ การทำงานของระบบส่วนที่ได้ออกแบบและพัฒนาขึ้น โดยแสดงเป็นแผนภาพ BPMN และแสดง Interface ของหน้าต่างๆ ในระบบ

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการเตรียมการทดลองทั้งการจัดเตรียมฮาร์ดแวร์ ซอฟต์แวร์ สภาวะแวดล้อมในการทำการทดลอง ข้อมูลทดสอบ การทำงานหรือการจำลองการทำงานของระบบ ผลการทดลอง ค่าสมรรถนะของระบบ การวัดประสิทธิภาพของระบบ การวิเคราะห์ผลการทดลองหรือผลการทำงานทั้งหมด

บทที่ 5 บทสรุป กล่าวถึงบทสรุปของโครงงาน วิจารณ์สิ่งที่ได้รับจากโครงงาน ข้อจำกัด รวมถึงปัญหาอุปสรรคต่างๆ ของโครงงาน และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ภาษา Python

2.1.1 ภาษา Python คือ

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ที่มีความสามารถสูง ถูกสร้างขึ้นมาโดย Guido Van Rossum ใน พ.ศ. 2533 โค้ดของ Python ถูกสร้างขึ้นมาจากภาษา C โดยทำงานเป็นภาษา Script การประมวลผลจะทำในแบบ Interpreter คือจะประมวลผลไปทีละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ

2.1.2 ข้อดีของภาษา Python

1) Source Code อ่านเข้าใจได้ง่าย ทำให้ง่ายต่อการเรียนรู้
2) ใช้ได้หลาย Platform
3) สนับสนุนทั้งการเขียนโปรแกรมเชิงวัตถุ และโครงสร้าง
4) จัดการ Memory ให้โดยอัตโนมัติ
5) จัดการตัวแปรให้ตามชนิดของข้อมูลที่บรรจุอยู่โดยอัตโนมัติ

2.1.3 การนำมาใช้ในชิ้นงาน

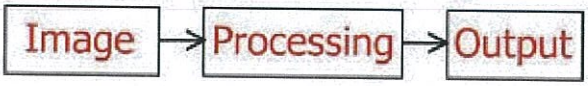
ใช้เป็นภาษาหลักในการพัฒนาชิ้นงาน ทั้งส่วนของส่วนตรวจจับ และส่วนของตัวเกม

2.2 Image Processing

2.2.1 Image Processing คือ

Computer vision คือระบบการมองเห็น หรือระบบการมองภาพของคอมพิวเตอร์ เพื่อจะนำข้อมูลที่ได้นั้นมาประมวลผล และทำความเข้าใจ จดจำ แยกแยะ สิ่งต่างๆ เพื่อนำไปใช้ประโยชน์

การประมวลผลภาพ(Image processing) คือ การนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่เราต้องการทั้งในเชิงคุณภาพและปริมาณ



รูป 2.1 การทำงานของ Image Processing

โดยมีขั้นตอนต่างๆ ที่สำคัญ คือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่เราสนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนที่ของวัตถุในภาพ จากนั้นเราสามารถนำข้อมูลเชิงปริมาณเหล่านี้ไปวิเคราะห์ และสร้างเป็นระบบ เพื่อใช้ประโยชน์ในงานด้านต่างๆ

การทำงานลักษณะนี้เป็นการประยุกต์ใช้งานการประมวลผลสัญญาณบนสัญญาณ2 มิติ เช่น ภาพนิ่ง หรือภาพวีดิทัศน์ และยังรวมถึงสัญญาณ 2 มิติอื่นๆ ที่ไม่ใช่ภาพด้วย

แนวความคิดและเทคนิค ในการประมวลผลสัญญาณ สำหรับสัญญาณ 1 มิตินั้น สามารถปรับมาใช้กับภาพได้ไม่ยาก แต่นอกเหนือจากเทคนิคจากการประมวลผลสัญญาณแล้ว การประมวลผลภาพก็มีเทคนิคและแนวความคิดที่เฉพาะ (เช่น Connectivity และ Rotation Invariance) ซึ่งจะมีความหมายกับสัญญาณ 2 มิติเท่านั้น แต่อย่างไรก็ตามเทคนิคบางอย่าง จากการประมวลผลสัญญาณใน 1 มิติ จะค่อนข้างซับซ้อนเมื่อนำมาใช้กับ 2 มิติ

ในช่วงแรกๆ การประมวลผลภาพนั้น จะอยู่ในรูปของการประมวลผลสัญญาณ Analog โดยใช้อุปกรณ์ปรับแต่งแสง (Optics) ซึ่งวิธีเหล่านั้นก็ไม่ได้หายสาบสูญ หรือเลิกใช้ไป ยังมีใช้เป็นส่วนสำคัญ สำหรับการประยุกต์ใช้งานบางอย่าง เช่น ฮอโลกราฟี (Holography) แต่เนื่องจากอุปกรณ์คอมพิวเตอร์ในปัจจุบันราคาถูกลง และเร็วขึ้นมาก การประมวลผลภาพดิจิทัล (Digital Image Processing) จึงได้รับความนิยมมากกว่า เพราะการประมวลผลที่ทำได้ซับซ้อนขึ้น แม่นยำ และง่ายในการลงมือปฏิบัติ

2.2.2 การนำมาใช้ในชิ้นงาน

ใช้ในการตรวจจับจอเกมที่ถูกฉายโดยเครื่องฉาย ลูกบอล และ Laser Pointer จากภาพที่ได้รับมาจากกล้อง

2.3 Library OpenCV

2.3.1 OpenCV คือ

OpenCV เป็น Library แบบเปิดเผยรหัส (Library Open Source) ถูกพัฒนาขึ้นโดยได้รับการสนับสนุนจาก Intel Corporation จำกัด ใช้ในภาษา C++ และ Python สำหรับการพัฒนาโปรแกรมที่เกี่ยวข้องกับ Image Processing และ Computer Vision โดยสามารถพัฒนาได้ทั้งในระบบปฏิบัติการ Windows และ ระบบปฏิบัติการ Linux

2.3.2 ฟังก์ชั่นที่นำมาใช้

2.3.2.1 HoughCircle

ใช้ในการค้นหาวงกลมภายในภาพเฉดสีเทาโดยใช้ Hough transform

การเรียกใช้ :

cv2.HoughCircles(image, method, dp, minDist[, circles[, param1[,param2[,minRadius[, maxRadius]]]]]) ⟶ circles

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) **Image** : 8-bit, Single-Channel, ภาพเฉดเทา
2) **circles** : ผลลัพธ์ของวงกลมที่เจอ โดยจะเก็บเป็นเวกเตอร์ที่มี 3 องค์ประกอบ คือ (x, y, Radius) โดยที่ x และ y เป็นตำแหน่งของจุดศูนย์กลางวงกลมในภาพ
3) **Method** : ตัวเลือกกระบวนการที่จะใช้ในการหาวงกลม ในปัจจุบันมีเพียง Method เดียวคือ CV_HOUGH_GRADIENT
4) **Dp** : คือตัวหารความละเอียดของภาพที่นำมาใช้ในฟังก์ชั่น เช่น dp = 1 จะใช้ภาพความละเอียดเท่าภาพจริง ถ้า dp = 2 จะใช้ความละเอียดเท่ากับครึ่งหนึ่งของภาพจริง
5) **minDist** : ระยะห่างที่น้อยที่สุดระหว่างจุดศูนย์กลางของวงกลมแต่ละวงที่หาเจอ
6) **Param1** : Canny Threshold ของ Hough Gradient
7) **Param2** : Accumulator Threshold ของ Hough Gradient
8) **minRadius** : ขนาดที่เล็กที่สุดของวงกลมที่ต้องการหา
9) **maxRadius** : ขนาดที่ใหญ่ที่สุดของวงกลมที่ต้องการหา

2.3.2.2 InRange

ใช้ในการเลือกค่าสีที่ต้องการ

การเรียกใช้ : cv2.inRange(src, lowerb, upperb[, dst]) ⟶dst

1) **Src** : รูปที่จะนำมาใช้เลือก
2) **Lowerb** : ช่วงค่าสี HSV ฝั่งน้อย
3) **Upperb** :ช่วงค่าสี HSVฝั่งมาก
4) **Dst** : รูปผลลัพธ์ของพื้นที่ๆ สนใจเป็นรูปสีขาวดำมีชนิดเป็น CV_8U และขนาดเท่ากับรูปที่นำมาใช้เลือก

2.3.2.3 Canny

ใช้ในการหาขอบของเส้นในรูป

การเรียกใช้ : cv2.Canny(image, threshold1, threshold2[, edges[, apertureSize[, L2gradient]]])

2.3.2.4 Contour

ใช้ในการหารูปร่างของเส้นขอบที่ได้มาจากการใช้ Canny เพื่อใช้สำหรับหาตำแหน่งของจอที่ฉายจากเครื่องฉาย

การเรียกใช้ : cv2.findContours(image, mode, method[, contours[, hierarchy[, offset]]])

2.3.3 การนำมาใช้ในชิ้นงาน

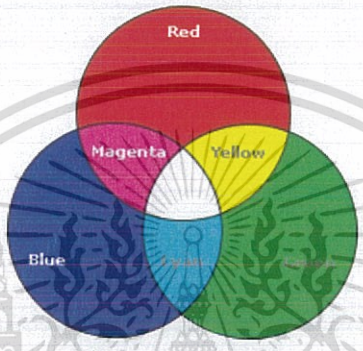
ใช้ในการทำ Image Processing เพื่อพัฒนาการตรวจจับจอเกมที่ถูกฉายโดยเครื่องฉาย ลูกบอล และ Laser Pointer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบสี RGB และ HSV

2.4.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่ประกอบด้วยแม่สี 3 สี คือแดง (Red) เขียว (Green) และน้ำเงิน (Blue) เมื่อนำมาผสมผสานกันทำให้เกิดสีต่างๆ บนจอคอมพิวเตอร์มากถึง 16.7 ล้านสี ซึ่งใกล้เคียงกับสีที่ตาเรามองเห็นปกติ สีที่ได้จากการผสมสีขึ้นอยู่กับความเข้มของสี โดยถ้าสีมีความเข้มมาก เมื่อนำมาผสมกันจะทำให้เกิดเป็นสีขาว จึงเรียกระบบสีนี้ว่าแบบ Additive หรือการผสมสีแบบบวก



รูป 2.2 ระบบสี RGB

2.4.2 ระบบสี HSV

ระบบสี HSV (Hue Saturation Value) ใช้หลักการแยกความสว่างออกจากเนื้อสีของจุดภาพ โดยปริภูมิสี HSV นั้นแทนค่าด้วยเวกเตอร์สามมิติ ซึ่งประกอบด้วย H แทนค่าเนื้อสี (Hue) S แทนค่าความอิ่มตัวของสี (Saturation) และ V แทนค่าความสว่างของแสง (Value) โดย Hue คือค่าสีของสีหลัก (แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดงเท่ากับ 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา

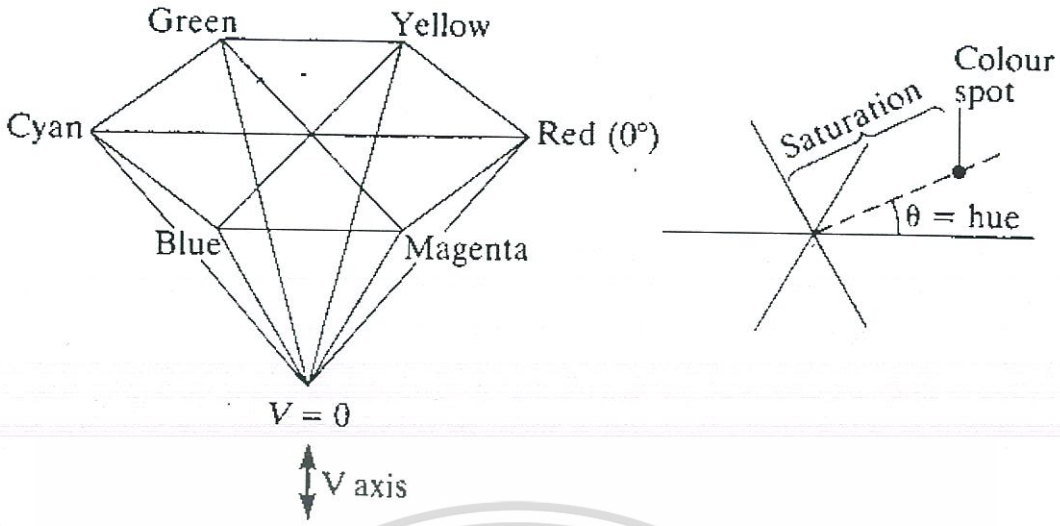
Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$red_h = red - min(red, green, blue) \tag{2.1}$$

$$green_h = green - min(red, green, blue) \tag{2.2}$$

$$blue_h = blue - min(red, green, blue) \tag{2.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.3 ระบบสี HSV

จากลักษณะ Model ของระบบ Hue พบว่า จะมีค่าอย่างน้อยหนึ่งค่าเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 Hue จะเป็นค่าสีที่มีค่าเป็นไปตามสีที่สาม และถ้าทั้งสามสีมีค่าเท่ากับ 0 จะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ

การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0 ดังสมการ 2.4

$$\frac{(240 \times blue_h) + (120 \times green_h)}{blue_h + green_h} \tag{2.4}$$

Saturation คือความบริสุทธิ์ของสี ซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วน แต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย

Saturation สามารถคำนวณได้ดังสมการ 2.5

$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)} \tag{2.5}$$

Value คือความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จากสมการ 2.6

$$value = \max(red, green, blue) \tag{2.6}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 การนำมาใช้ในชิ้นงาน

เป็นทฤษฎีที่นำมาประยุกต์ใช้ในการตรวจจับลูกบอล

2.5 Edge Detection

2.5.1 การค้นหาขอบภาพ (Edge Detection)

หลักการคือ ประมวลผลสัญญาณบนสัญญาณ 2 มิติ เช่น ภาพนิ่ง หรือภาพเคลื่อนไหว รวมถึงสัญญาณ 2 มิติอื่นๆ ที่ไม่ใช่ภาพ การแยกวัตถุ 2 วัตถุออกจากกันหรือแยกออกจากพื้นหลัง (Background) ต้องใช้การค้นหาขอบวัตถุที่อยู่ในภาพนั้น การค้นหาขอบภาพคือการหาเส้นรอบรูปที่เกิดจากความสว่างของภาพที่เปลี่ยนไปทันที โดยดูจากขนาดหรืออัตราการเปลี่ยนขนาด การค้นหาขอบภาพเป็นการดึงส่วนประกอบของภาพที่มีจุดเด่นออกจากพื้นหลัง มีวิธีการแยกได้เป็นสองลักษณะคือการค้นหาขอบภาพโดยใช้อนุพันธ์อันดับหนึ่ง (Gradient Method) และการค้นหาขอบภาพโดยใช้อนุพันธ์อันดับสอง (Laplacian Method)

2.5.2 การค้นหาขอบภาพโดยใช้อนุพันธ์อันดับที่หนึ่ง

วิธีการค้นหาขอบภาพโดยใช้อนุพันธ์อันดับหนึ่ง (Gradient Operator:∇) มีหลักการคือ บริเวณขอบของวัตถุในภาพ (เมื่อ P(x, y) คือฟังก์ชั่นของภาพ) จะมีค่า Gradient ที่สูง การพิจารณาขนาดของ Gradient (Gradient Magnitude ($|\nabla P|$)) เปรียบเทียบกับค่าอ้างอิง (Threshold) ที่กำหนดขึ้น เมื่อค่าของ $|\nabla P|$ มีค่ามากว่าค่าอ้างอิง แสดงว่าจุดดังกล่าวคือขอบของวัตถุที่ปรากฏในภาพที่จุด P(x, y) การค้นหาขอบของวัตถุโดยใช้อนุพันธ์อันดับหนึ่ง เป็นวิธีแยกส่วนประกอบของภาพและเมื่อความไม่ต่อเนื่องของค่า Pixels บริเวณรอยต่อระหว่างวัตถุกับพื้นหลังและค่าอนุพันธ์ย่อยที่ไม่ต่อเนื่องตามทิศทางของ Gradient ของแนวแกน x และแกน y กำหนดค่าได้ตามสมการ 2.7 และ 2.8

$$\nabla_x P(x,y) = P(x,y) - P(x-1,y) \tag{2.7}$$

$$\nabla_y P(x,y) = P(x,y) - P(x,y-1) \tag{2.8}$$

ขนาดของ Gradient ของ P(x,y) กำหนดค่าได้จากสมการ 2.9

$$|\nabla P(x,y)| = \sqrt{\left(\nabla_x P(x,y)\right)^2 - \left(\nabla_y P(x,y)\right)^2} \tag{2.9}$$

และเพื่อให้ง่ายต่อการคำนวณ ประมาณค่าขนาดของ Gradient ได้ตามสมการ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$|\nabla P(x,y)| = \left|\left(\nabla_x P(x,y)\right)\right| \left|\left(\nabla_y P(x,y)\right)\right| \tag{2.10}$$

การค้นหาขอบของภาพที่มีองค์ประกอบของเส้นตรงด้วยวิธี Canny Edge Detection เป็นวิธีแบ่งออกเป็นส่วนย่อยในแต่ละ Pixel กำหนดรูปแบบการเปลี่ยนแปลงภาพให้มีความเรียบด้วยตัวกรอง Gaussian ก่อนคำนวณหาขนาดและทิศทางของ Gradient ขึ้นมาและค่าของ Mask กำหนดค่าได้ดังสมการ 2.11 และ 2.12

$$Mask(E_x) = \begin{bmatrix} Z_{x1} & Z_{x2} & Z_{x3} \\ Z_{x4} & Z_{x5} & Z_{x6} \\ Z_{x7} & Z_{x8} & Z_{x9} \end{bmatrix} = E_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.11}$$

$$Mask(E_y) = \begin{bmatrix} Z_{y1} & Z_{y2} & Z_{y3} \\ Z_{y4} & Z_{y5} & Z_{y6} \\ Z_{y7} & Z_{y8} & Z_{y9} \end{bmatrix} = E_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.12}$$

กำหนดให้ขนาดของ Mask เท่ากับ 3×3 มีค่าเท่ากับ E_x และ E_y หาค่าอนุพันธ์อันดับหนึ่งของ $\frac{\partial P}{\partial x}$ และ $\frac{\partial P}{\partial y}$ ขนาดและทิศทางของ Gradient หาได้จากสมการ 2.13 และ 2.14

$$M(i,j) = \sqrt{E_x^2(i,j) + E_y^2(i,j)} \tag{2.13}$$

$$\theta(i,j) = \tan^{-1}\left(E_x^2(i,j) + E_y^2(i,j)\right) \tag{2.14}$$

การกำหนดค่า Mask ของแนวแกน x และ y ทำให้ขนาดและทิศทางของ Gradient เปลี่ยนแปลง ทำให้ลดขนาดของขอบในแต่ละ Pixel ของขอบภาพได้ Pixel รอบข้างของทิศทางของ Gradient ใดที่ไม่ใช่ค่าสูงสุดจะไม่ถือว่าเป็นขอบ ดังนั้นจะมีเพียง Pixel หนึ่งเดียวที่เป็นขอบของวัตถุในภาพ

2.5.3 การค้นหาขอบภาพโดยใช้อนุพันธ์อันดับสอง

การค้นหาขอบภาพโดยใช้อนุพันธ์อันดับสอง เปรียบได้กับการหาอนุพันธ์อันดับสองของภาพเพื่อให้ได้ขอบของภาพ โดยภาพที่ผ่านการหาอนุพันธ์อันดับสอง (Laplacian Operator ($\nabla^2 P$)) บริเวณที่เป็นส่วนของขอบจะเด่นชัดขึ้น การประมาณค่าของอนุพันธ์อันดับสอง โดยใช้ Mask ของสมการ 2.11 และ 2.12 ทำให้ตำแหน่งของบริเวณขอบของวัตถุในภาพคือค่าจุดผ่านศูนย์ของอนุพันธ์อันดับสองนั่นเอง โดยการหาอนุพันธ์อันดับสองสามารถหาได้จากสมการ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\nabla^2 P = \frac{\partial^2 \mathrm{P}}{\partial x^2} + \frac{\partial^2 \mathrm{P}}{\partial \mathrm{y}^2} \tag{2.15}$$

การค้นหาขอบของภาพด้วยอนุพันธ์อันดับสอง บริเวณขอบจะมีตำแหน่งเดียวกับค่าผ่านศูนย์ของค่าจากอนุพันธ์อันดับสอง พิจารณาได้จากตำแหน่งที่ Pixel เปลี่ยนแปลงจากค่าที่เป็นบวกเป็นค่าลบหรือจากค่าที่เป็นลบไปเป็นค่าที่เป็นบวก ซึ่งการหาขอบโดยใช้อนุพันธ์อันดับสอง ไม่สนใจทิศทางของภาพในแนวแกน x และ แกน y และกำหนดจุดที่ค่า y เป็นจุดผ่านศูนย์ วิธีนี้ใช้เวลาในการคำนวณมากกว่าการค้นหาขอบโดยใช้ อนุพันธ์อันดับหนึ่ง กล่าวคือเราสามารถตรวจจับความไม่ต่อเนื่องของ Pixel ใน Domain รูปภาพ ได้โดยใช้อนุพันธ์ของภาพนั้นเอง

2.5.4 การค้นหาขอบภาพด้วย Canny

ขั้นตอนการค้นหาขอบภาพโดยวิธีของ Canny ประกอบด้วย 4 ขั้นตอน เริ่มต้นจากการปรับภาพให้เรียบด้วยตัวกรอง Gaussian เพื่อกำจัดสัญญาณรบกวน หลังจากนั้นหาอนุพันธ์อันดับหนึ่ง คำนวณค่าขนาดและทิศทางของ Gradient นำค่าที่ได้มาคำนวณค่าของ Non-maxima Suppression กับค่าขนาดของ Gradient เพื่อทำให้ได้ขอบที่บางลง ขั้นตอนสุดท้าย ใช้การกำหนดจุดอ้างอิงสองระดับ (Double Thresholding) เพื่อระบุค่าของ Pixel ที่เป็นขอบและช่วยเชื่อมต่อขอบ โดยในแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

2.5.4.1 การปรับภาพให้เรียบ

ขั้นตอนแรกของการค้นหาขอบด้วย Algorithm ของ Canny ต้องกำจัดสัญญาณรบกวนออกก่อน ด้วยวิธีการใช้ตัวกรอง Gaussian กำหนดกรอบเป็น Metrix ขนาด 3x3 หรือมีขนาดเท่ากับ 9 Pixels ในการกำหนดขนาดของตัวกรองนั้น หากมีขนาด Gaussian กว้างมาก จะลดสัญญาณรบกวนได้มาก แต่หากขนาดกรอบกว้างมากเกินไป จะมีผลให้ขอบย่อยๆ ที่เป็นส่วนรายละเอียดหายไป ผลของภาพที่ผ่านการปรับภาพให้เรียบด้วยตัวกรอง Gaussian หาได้จากสมการ 2.16

$$S_{(i,j)} = G_{(i,j,\sigma)} \cdot I_{(i,j)} \tag{2.16}$$

กำหนดให้

$I_{(i,j)}$ คือภาพที่ต้องการหาขอบ

$G_{(i,j,\sigma)}$ คือ Gaussian Smoothing Filter

σ คือค่าควบคุมระดับของการ Smoothing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.2 การคำนวณค่า Gradient

ขั้นแรกปรับภาพ $I_{(i,j)}$ ให้มีความเรียบ ผลลัพธ์ที่ได้คือค่าของภาพในฟังก์ชั่น $S_{(i,j)}$ ต่อมาหาค่าของ Gradient ในทิศทางของแกน x และแกน y และกำหนดขนาดของอนุพันธ์อันดับหนึ่งของ $Px_{(i,j)}$ และ $Qy_{(i,j)}$ ตามลำดับ ดังสมการ 2.17 และ 2.18

$$Px_{(i,j)} = \frac{\left(S_{(i,j+1)} - S_{(i,j)} + S_{(i+1,j+1)} - S_{(i+1,j)}\right)}{2} \tag{2.17}$$

$$Qy_{(i,j)} = \frac{\left(S_{(i,j)} - S_{(i+1,j)} + S_{(i,j+1)} - S_{(i+1,j+1)}\right)}{2} \tag{2.18}$$

นำค่า $Px_{(i,j)}$ และ $Qy_{(i,j)}$ ที่ผ่านการหาอนุพันธ์อันดับหนึ่ง เมื่อคำนวณการแปลงรูปแบบจากระนาบของระบบพิกัดฉากไปเป็นระนาบพิกัดเชิงขั้ว เพื่อหาขนาดและทิศทางของ Gradient โดยแทนค่าตามสมการ 2.17 และ 2.18 ได้ค่าขนาด Gradient ดังสมการ 2.19 และทิศทางของเกรเดียนต์ (Gradient Orientation) เท่ากับสมการ 2.19 และสามารถหาค่ามุม θ ออกมาได้เมื่อแทนค่าตัวแปรในสมการ 2.21

$$\mathrm{M(i,j)} = \sqrt{P_x{}^2(i,j) + Q_y{}^2(i,j)} \tag{2.19}$$

$$\theta(\mathrm{i,j}) = \tan^{-1}\left(P_x{}^2(i,j) + Q_y{}^2(i,j)\right) \tag{2.20}$$

$$\theta = \tan^{-1}(x,y) \tag{2.21}$$

2.5.4.3 Non-Maxima Suppression

การค้นหาขอบภาพด้วยวิธีการของ Canny นั้น จุดที่ถือเป็นเส้นขอบของภาพได้ต้องเป็นจุดที่ให้ค่าสูงสุดเฉพาะที่และเป็นทิศทางเดียวกับ Gradient การค้นหาขอบภาพโดยใช้อนุพันธ์อันดับหนึ่งทำให้ได้ขอบที่บางเพียง 1 Pixel ภาพที่ได้หลังการทำ Non-Maxima Suppression จะให้ค่าเป็นศูนย์ในทุกจุดยกเว้นจุดที่เป็น Local Maxima Point ซึ่งจะยังคงค่าเดิมไว้

2.5.4.4 Thresholding

แม้ว่าภาพจะผ่านการปรับให้เรียบในขั้นตอนแรกแล้วก็ตาม ภาพที่ได้อาจยังมีเส้นขอบที่ไม่ใช่ขอบจริงปรากฏอยู่ อันเนื่องจากสัญญาณรบกวนหรือลักษณะของวัตถุในภาพเป็นพื้นผิวที่มีลวดลายหรือมีรายละเอียดภายในมาก เพื่อลดปัญหาดังกล่าว จึงมีการกำหนดค่า Threshold ขึ้นมา 2 ค่า คือ High Threshold (T_1) และ Low Threshold (T_2) โดย Pixel ที่มีค่ามากกว่า T_1 จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปรับเป็น '1' เป็น Pixel ที่เป็นขอบ แต่ถ้าน้อยกว่า T_2 จะถูกปรับเป็น '0' ส่วนค่าที่อยู่ระหว่างค่า Threshold ทั้งสอง จะปรับเป็นค่า '0' หรือ '1' นั้นขึ้นอยู่กับ Pixel ที่อยู่รอบข้าง หากพบว่า Pixel ที่อยู่รอบข้าง Pixel ที่เป็นขอบ (ขอบมีค่า > T_1) มีค่ามากกว่า T_2 แล้ว จะปรับค่า Pixel ดังกล่าวให้มีค่าเป็น '1' และถือเป็นสมาชิกหนึ่งในขอบภาพด้วยเช่นกัน ดังนั้นการทำ Threshold จะทำให้ภาพที่มีขอบหนาขึ้นหรือบางลงนั่นเอง

2.5.5 การนำมาใช้ในชิ้นงาน

ใช้ในการตรวจจับหน้าจอที่ฉายจากเครื่องฉาย โดยดูจากรูปทรงของภาพที่ได้มาจากกล้อง ว่าตรงกับรูปทรงของขอบจอหรือไม่

2.6 Multi-Threads

2.6.1 Thread คือ

หน่วยการทำงานย่อยที่อยู่ใน Process มีการแบ่งปันทรัพยากรต่างๆ ใน Process นั้นๆ โดยปกติ Process ที่มี 1 Thread จะเรียกว่า Single thread หรือเรียกว่า Heavy Weight Process ซึ่งมักพบในระบบปฏิบัติการคอมพิวเตอร์รุ่นเก่า แต่ถ้า 1 Process มีหลาย Thread จะเรียกว่า Multithread หรือ Light Weight Process ซึ่งพบได้ในระบบปฏิบัติการคอมพิวเตอร์รุ่นใหม่ที่ใช้กันในปัจจุบันทั่วไป และ Multithread ก็เป็นที่นิยมมากกว่า Single Thread เหตุที่ต้องมี Thread คือการเรียกใช้หน่วยประมวลผลให้เกิดประโยชน์สูงสุด Thread ทำให้การทำงานของโปรแกรมง่าย มีประสิทธิภาพมากขึ้นและมีประโยชน์ต่อระบบที่มีหน่วยประมวลผลแบบ Multi-Core เพราะสามารถเรียกใช้ Thread หลายตัวได้พร้อมๆ กัน โดย Thread แต่ละตัว ของ Process เดียวกัน จะทำงานแตกต่างกันแต่มีความเกี่ยวข้องกันบางอย่างและต้องทำงานอยู่ภายใต้สภาพแวดล้อมเดียวกัน

2.6.2 ประโยชน์ของ Multi-Threads

2.6.2.1 การตอบสนอง

ในด้านการทำงาน มีการตอบสนองที่ดีกับผู้ใช้ ถ้าการทำงานของโปรแกรมประยุกต์ของผู้ใช้นั้นมีบางส่วนภายใน Process ถูก Block หรือใช้เวลามากเกินไป ระบบปฏิบัติการก็ยังสามารถจัดสรรให้งานอื่นๆ ภายใน Process นั้นประมวลผลต่อไปได้

2.6.2.2 การใช้ทรัพยากรร่วมกัน

สามารถใช้ Code, Application และ Memory ร่วมกันระหว่าง Process เดียวกันได้

2.6.2.3 ประหยัด

ประหยัดการใช้หน่วยความจำในการทำงานของ Process เนื่องจากแต่ละ Thread มีการใช้หน่วยความจำของ Process ร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2.4 ด้านโครงสร้างของ Multi-Thread

การเอื้อประโยชน์ด้านโครงสร้างระบบ ให้งานย่อยต่างๆ ภายใน Process สามารถทำงานร่วมกัน และใช้ทรัพยากรของ Process ร่วมกันได้

2.6.3 การนำมาใช้ในชิ้นงาน

ใช้แบ่งการทำงานของตัวเกมและส่วนตรวจจับออกจากกัน เพื่อให้ส่วนตรวจจับสามารถทำงานได้อย่างเต็มประสิทธิภาพ เนื่องจากสามารถทำงานไปพร้อมกันได้ทั้งสองส่วน

2.7 Pyglet Library

2.7.1 Pyglet Library คือ

Python Game Library โดยเป็น Open Source ที่เรียกใช้ OpenGL รองรับการใช้งานร่วมกับ Mouse และ Keyboard แสดงข้อความ รูปภาพ และเล่นสื่อ Multimedia มีฟังก์ชั่นที่หลากหลายเพียงพอในการเขียนเกมทั้งสองมิติและสามมิติ

2.7.2 ข้อดีของ Pyglet Library

1) รองรับการแสดงผลแบบเต็มหน้าจอ
2) นำมาใช้พัฒนาเกมได้ง่ายด้วยการเขียนแบบ Object-Oriented Programming
3) สนับสนุนการแสดงผลหลายรูปแบบทั้งภาพและเสียง
4) สนับสนุน Input หลายอย่าง และสามารถกำหนดวิธีการ Input ข้อมูลเองได้

2.7.3 การนำมาใช้ในชิ้นงาน

นำมาใช้ในการพัฒนาตัวเกมเบื้องต้น ที่ใช้กับส่วนตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

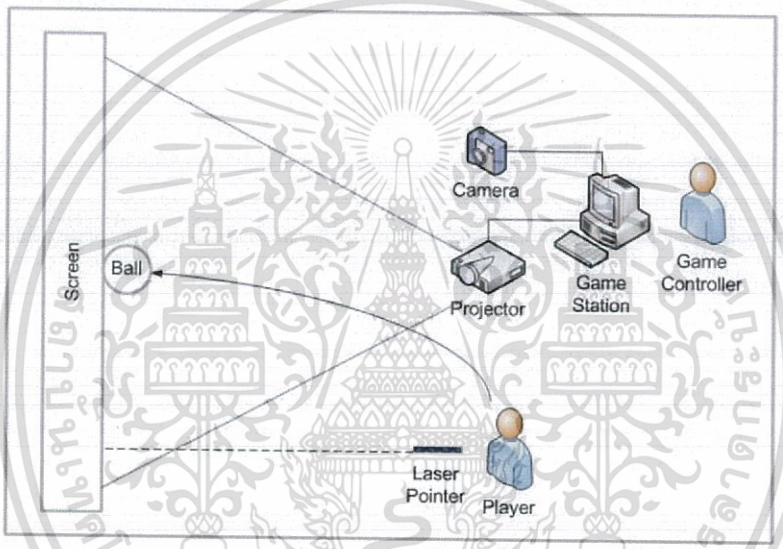
การออกแบบและการพัฒนา

3.1 รายละเอียดของระบบ

รายละเอียดของระบบแต่ละส่วนมีการทำงานดังนี้

3.1.1 โครงสร้างของระบบ

ระบบจะฉายภาพเกมไปยังผนัง ซึ่งเกมต่างๆ ที่ฉายมีลักษณะเป็นเกมยิง และผู้เล่นจะเล่นโดยการโยนลูกบอลกระทบผนังแทนการยิง โดยโครงสร้างของอุปกรณ์ในระบบเป็นดังรูป



รูป 3.1 โครงสร้างของอุปกรณ์ในระบบ

อุปกรณ์ที่เป็นส่วนประกอบของเครื่องเล่นเกมมีดังนี้

1) **Projector** (เครื่องฉาย Projector)
2) **Screen** (ผนัง) ซึ่งเปรียบเสมือนจอภาพ สำหรับฉายภาพเกม
3) **Camera** (กล้อง) ใช้ในการตรวจจับลูกบอลที่เกิดขึ้นด้านหน้าผนัง
4) **Game Station** ส่วนประมวลผลของเกม
5) **Game Controller** ผู้ควบคุมเครื่องเล่นเกม

อุปกรณ์ที่ใช้ในการควบคุมเกมมีดังนี้

1) **Player** (ผู้เล่น)
2) **Ball** (ลูกบอล) ปาใส่ผนังเพื่อใช้แทนการยิงในเกม
3) **Laser pointer** ใช้ในการปรับตั้งค่าต่างๆ ของระบบ
4) **Game Station** ส่วนประมวลผลของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ความสามารถของระบบ

1) เปิด/ปิด การทำงานของลูกบอล
2) เปิด/ปิด การทำงานของ Laser Pointer
3) สามารถเลือกเกมที่จะเล่นได้
4) ตั้งค่าต่างๆ ของเกมในขณะที่เล่นอยู่ได้ โดยใช้ Laser Pointer
5) เมื่อมีการคลิกซ้ายที่หน้าจอ จะเหมือนมีลูกบอลกระทบผนังจุดที่คลิก
6) เมื่อมีการคลิกขวาที่หน้าจอ จะเหมือนมี Laser Pointer ชี้ที่ผนังจุดที่คลิก

3.1.3 ส่วนของเกม

รายการเกมในเบื้องต้นคือ

3.1.3.1 เกมพื้นฐาน

สำหรับทดสอบระบบ โดยเมื่อโยนลูกบอลกระทบผนังจะเกิดรูปการ์ตูนขึ้นบนผนัง เกมนี้สามารถตั้งค่าล้างหน้าจอโดยใช้ Laser Pointer ได้

3.1.3.2 เกมปาเป้า

จะมีรูปเป้าขึ้นที่จอภาพบนผนัง และเมื่อโยนลูกบอลกระทบผนังตามจำนวนครั้งที่กำหนด จะนับคะแนนให้ผู้เล่น Laser Pointer สามารถตั้งค่าขนาดของเป้าได้ เปลี่ยนแปลงการตั้งค่าจำนวนการโยนลูกบอลได้

3.1.3.3 เกมปาแมลงวัน

จะมีรูปแมลงวัน จำนวนตามที่ตั้งค่าไว้ในระบบ บินด้วยความเร็วแบบสุ่มบนผนัง เมื่อโยนลูกบอลโดนแมลงวัน รูปแมลงวันจะหายไปและจะนับคะแนน ตัวชี้เลเซอร์สามารถตั้งค่าความเร็วมากที่สุดของแมลงวัน จำนวนแมลงวัน เปิดปิดการนับเวลา เปลี่ยนแปลงการตั้งค่าเวลาในเกมได้

3.1.4 ข้อจำกัดของระบบ

1) ลูกบอลที่ใช้เล่นเกม ใช้ได้เฉพาะสีที่ได้กำหนดไว้เท่านั้น
2) พื้นหลังของผนังที่ใช้เล่นต้องเป็นสีขาวเท่านั้น
3) หากลูกบอลเคลื่อนที่ด้วยความเร็วมาก อาจเกิดการผิดพลาดในการตรวจจับลูกบอลได้

3.2 เครื่องมือที่ใช้พัฒนา

3.2.1 สภาพแวดล้อมในการพัฒนา

1) **หน่วยประมวลผล** Intel® Core™ i5 CPU M450 Dual-Core Processor 2.4 GHz
2) **หน่วยความจำหลัก** 4 GB
3) **ระบบปฏิบัติการ** Windows 10 64-bit
4) **Hard Disk** Seagate Momentus ST9500420AS 500GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

5) กราฟิกการ์ด NVIDIA GeForce GT 335M

6) กล้อง Logitech Pro HD Webcam C920

3.2.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา

1) Python Version 2.7.5

2) OpenCV Version 2.4.11

3) Numpy Version 1.7.1

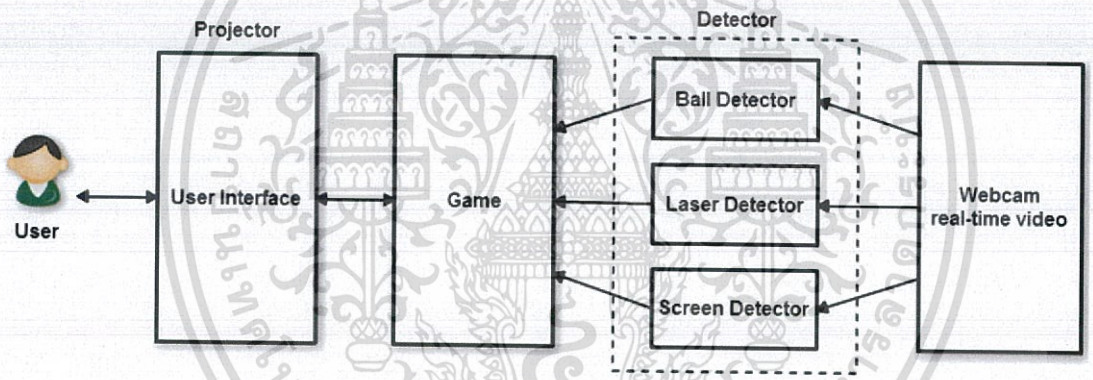
4) Pycharm Community Edition Version 4.5

5) Pyglet Version 1.2.4

3.2.3 ภาษาที่ใช้ในการพัฒนา

1) ภาษา Python

3.3 โครงสร้างของระบบ



รูป 3.2 System Architecture

ส่วนตรวจจับต่างๆ ทั้งบอล หน้าจอ และ Laser นั้นจะรับรูปภาพที่ได้จากการบันทึกวีดีโอจากกล้องมาทีละหนึ่งภาพ แล้วนำภาพที่ได้ไปประมวลผลว่ามีเหตุการณ์ใดที่เราสนใจเกิดขึ้นบ้างในภาพนั้น เมื่อมีเหตุการณ์ที่เราสนใจเกิดขึ้นจะทำการส่งข้อมูลตำแหน่งของเหตุการณ์นั้นๆ ไปยังตัวเกม โดยถ้าเป็นส่วนตรวจจับลูกบอลจะทำการส่งตำแหน่งที่ลูกบอลกระทบกับผนังไปให้ตัวเกม ส่วนตรวจจับ Laser จะทำงานเมื่อมี Laser ปรากฏขึ้นในภาพที่ได้จาก Webcam จึงส่งตำแหน่ง Laser ไปยังตัวเกมโดยจะส่งไปเมื่อ Laser ไปอยู่ในตำแหน่งที่กำหนดไว้ในหน้าจอเกมโดยทั้งส่วนตรวจจับบอลและ Laser จะทำงานตรวจจับอยู่เบื้องหลังตลอดเวลา ส่วนการตรวจจับหน้าจอจะทำเมื่อผู้ใช้กดปุ่มเพื่อสั่งให้ทำงาน เนื่องจากการตรวจจับหน้าจอจำเป็นจะต้องทำก็ต่อเมื่อเพิ่งเปิดเกมขึ้นมา หรือในกรณีที่กล้องหรือเครื่องฉายมีการขยับในระหว่างการเล่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
144367

ตัวเกมจะทำการแสดงผลหน้าจอเกมในปัจจุบันออกมาผ่านทางการฉายด้วยเครื่องฉายไปเรื่อยๆ แล้วรอจนกว่าจะมีการส่งค่ามาจากส่วนตรวจจับจึงจะนำค่าที่ได้ไปเลือกว่าจะเกิดเกตุการณ์อะไรขึ้นจากค่าดังกล่าว เช่นในเกมพื้นฐาน เมื่อได้รับค่าตำแหน่งของลูกบอลที่กระทบกับจอเกมมาจากส่วนตรวจจับ ตัวเกมก็จะทำการแสดงรูปการ์ตูนขึ้นมาบริเวณตำแหน่งนั้น

3.4 การออกแบบและผังการทำงานของระบบ

3.4.1 แผนภาพ BPMN

BPMN ย่อมาจาก Business Process Modeling Notation คือ Model แสดงกระบวนการทำงาน เพื่อแสดงถึงสิ่งที่เกิดขึ้น ขั้นตอน ข้อมูลที่ใช้ในการทำงาน และข้อมูลที่ระบบจัดเก็บ โดยแสดงออกมาเป็นแผนภาพ ซึ่งในโครงงานนี้ คณะผู้จัดทำนำแผนภาพ BPMN มาใช้แทนแผนภาพต่างๆ เช่น Flow Chart, Data Flow Diagram หรือ UML Activity Diagram

3.4.1.1 แผนภาพ BPMN ของหน้าเลือกเกม

เมื่อผู้เล่นต้องการเล่นเกม ต้องทำการติดตั้งอุปกรณ์ และเปิดการทำงานของระบบจะเข้ามาสู่หน้าแรก ซึ่งมีเมนูต่างๆ คือ

1) **แถบสำหรับเลือกเกม** จะมีรายชื่อเกมให้เลือก เมื่อเลือกแล้วจะมีรายละเอียดและวิธีการเล่นของเกมนั้นๆ แสดงบนช่องแสดงผลภาพ
2) **ช่องแสดงผลภาพ** แสดงรายละเอียดและวิธีการเล่นเกมที่เลือกไว้
3) **ตั้งค่าการเปิดปิดเสียง**
   a) เปิด/ปิด เสียงเพลงภายในหน้าเลือกเกม
   b) เปิด/ปิด เสียงกดปุ่มภายในหน้าเลือกเกม
4) **ปุ่ม Select** สำหรับเข้าไปยังเกมที่ผู้ใช้ได้เลือกไว้
5) **ปุ่ม Quit** สำหรับออกจากระบบ

อธิบายแผนภาพ BPMN การทำงานของระบบได้ดังนี้

1) ผู้ใช้ต้องการที่จะเล่นเกม
2) ติดตั้งและเชื่อมต่ออุปกรณ์ต่างๆ ในตำแหน่งที่เหมาะกับการเล่น เช่น พื้นที่เล่นเกมติดกำแพง มีพื้นที่พอสำหรับการโยนและเก็บลูกบอล
3) ผู้ใช้งานสามารถตั้งค่าเปิด/ปิดเสียงภายในหน้าเลือกเกม
4) ผู้ใช้เลือกเกมที่สนใจจากรายชื่อเกมบนแถบทางด้านขวาของหน้าจอ
5) กดเลือกเกม For Test และกดปุ่ม Select เพื่อเข้าสู่เกม For Test
6) แสดงหน้าเลือกเกม
7) แสดงรายละเอียดและวิธีเล่นของเกมที่ผู้ใช้เลือกไว้
8) แสดงหน้าเกม For Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) รอการติดตั้งระบบของผู้ใช้

10) การติดตั้งระบบเสร็จสิ้น

11) ระบบพร้อมที่จะเริ่มทำงาน

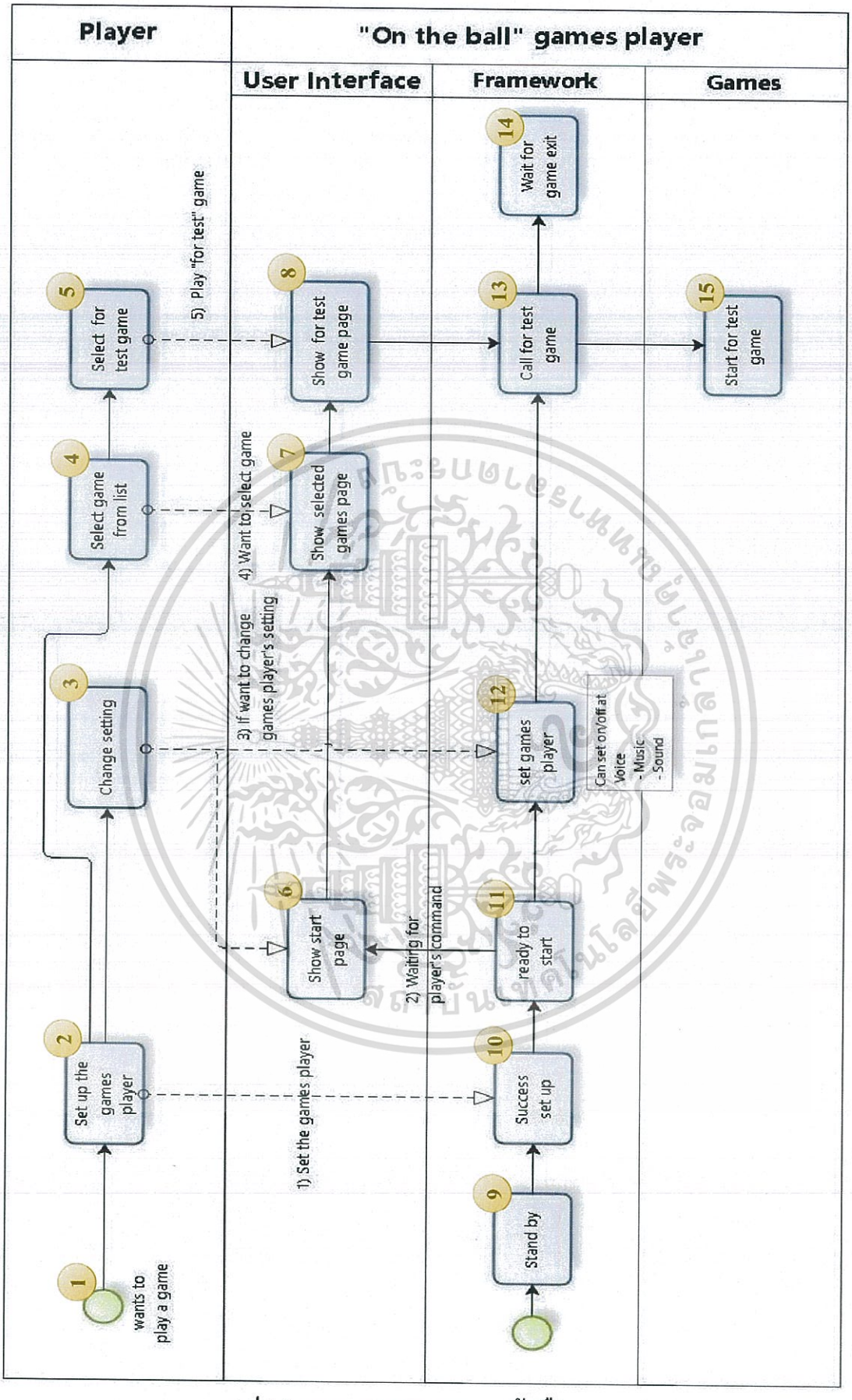
12) ส่วนตั้งค่าเสียงภายในหน้าเลือกเกม

13) ระบบเรียกเกม For Test ขึ้นมาทำงาน

14) รอจนกว่าผู้ใช้จะออกจากเกม

15) ตัวเกม For Test เริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.3 แผนภาพ BPMN ของหน้าเลือกเกม

3.4.1.2 ตัวอย่างการทำงานของเกม " For test "

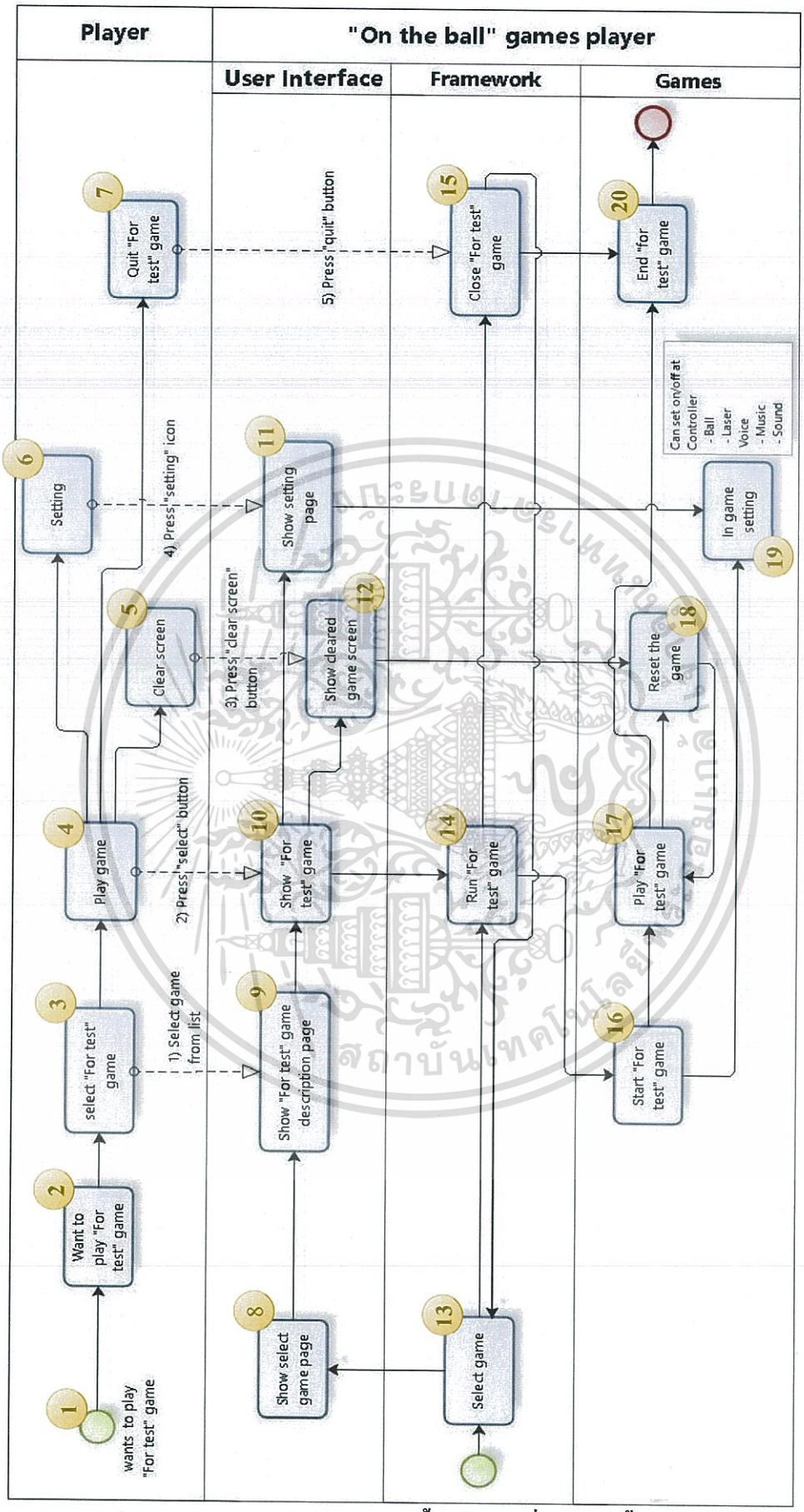
ในหน้าเมนูเลือกเกม หลังจากที่ผู้เล่นเลือกเล่นเกม For test ระบบจะแสดงหน้าที่มีคำอธิบายของเกม For Test ผู้เล่นสามารถเลือก

1) **ปุ่ม Select** เพื่อเข้าสู่เกม For Test ที่ผู้ใช้ได้เลือกไว้ จากนั้นผู้เล่นก็เริ่มโยนลูกบอลเพื่อเล่นเกมได้ เมนูที่ผู้เล่นสามารถเลือกได้ในขณะเล่นเกม คือ
   a) **ล้างหน้าจอ** ระบบจะลบรูปแมวบนจอเกม, Reset คะแนน และเริ่มเกมใหม่
   b) **ตั้งค่า** ตั้งค่าเปิดปิด การควบคุมเกมด้วยลูกบอลและ Laser Pointer และตั้งค่าเปิดปิดเสียงภายในเกม
   c) **ออกจากเกม** ระบบจะออกจากเกม และย้อนกลับไปหน้าเลือกเกม

อธิบายแผนภาพ BPMN การทำงานของเกมเบื้องต้นเกมที่ 1 เกมพื้นฐานได้ดังนี้

1) การทำงานของเกมแรก
2) ผู้ใช้ต้องการเล่นเกม For test
3) ผู้ใช้เลือกเกม For Test
4) ผู้ใช้เล่นเกม For Test
5) ผู้ใช้ต้องการล้างหน้าจอเกม
6) ผู้ใช้ต้องการตั้งค่าต่างๆภายในเกม
7) ผู้ใช้ต้องการออกจากเกม For test
8) แสดงหน้าเลือกเกม
9) แสดงคำอธิบายและวิธีการเล่นเกม For Test บนจอภาพ
10) แสดงหน้าตัวเกม For Test
11) แสดงหน้าสำหรับตั้งค่าเกม
12) แสดงหน้าเกมที่ถูกล้างหน้าจอแล้ว และพร้อมสำหรับการเริ่มเกมใหม่
13) เลือกเกม
14) เรียกเกม For Test ขึ้นมาทำงาน
15) ระบบปิดการทำงานของเกม For Test และกลับไปยังหน้าเลือกเกม
16) เกม For Test เริ่มทำงาน
17) ตัวเกม For Test กำลังทำงาน
18) ตัวเกมถูกล้างค่าต่างๆให้เป็นค่าเริ่มต้น
19) ตัวเกมอยู่ในระหว่างการตั้งค่าโดยผู้ใช้
20) ออกจากเกม For test

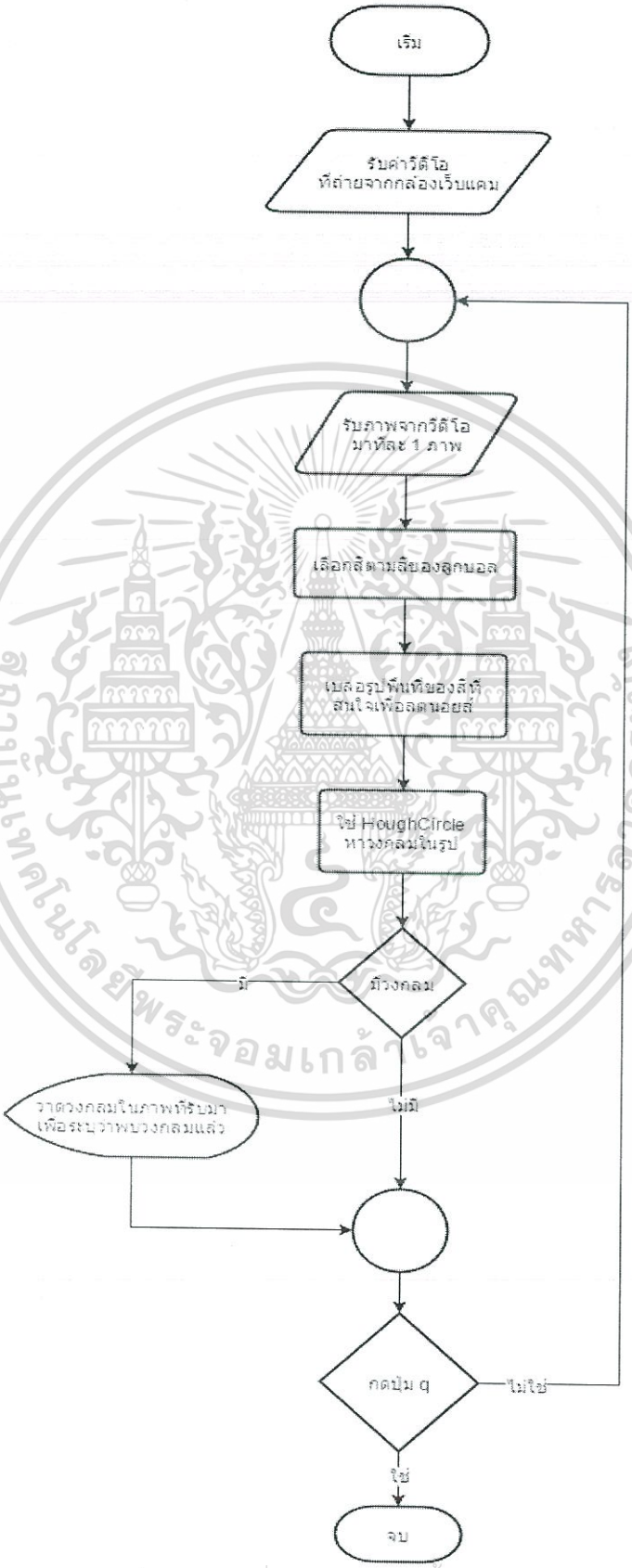
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.4 แผนภาพ BPMN ของเกมเบื้องต้นเกมที่ 1 “ เกมพื้นฐาน ”

3.5 ขั้นตอนการทำงานของระบบ

การตรวจจับลูกบอลของระบบ มีรูปแบบการทำงาน ตามแผนผังนี้



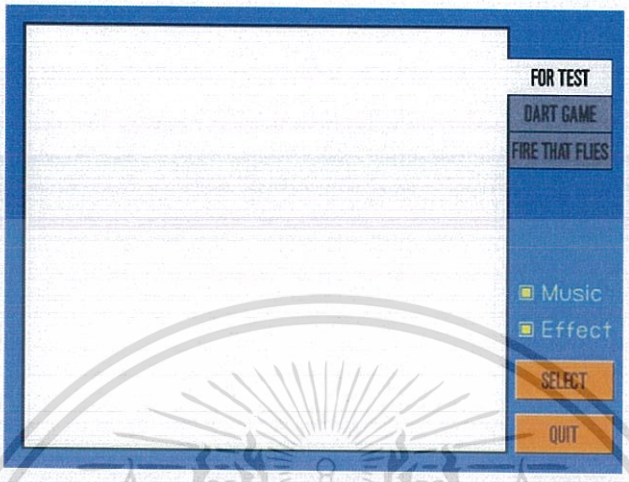
รูป 3.5 Flowchart ของการตรวจจับลูกบอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบ Interface ของเกม

ทางคณะผู้จัดทำได้ทำการออกแบบโครงสร้างของ Interface ต่างๆ ไว้ดังนี้

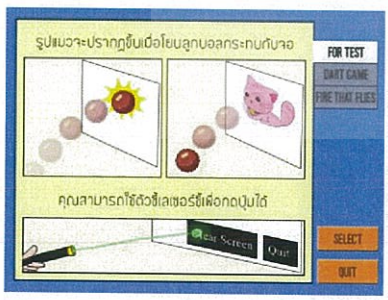
3.6.1 ตัวอย่างการแสดงผลของเกม



รูป 3.6 หน้าเลือกเกม

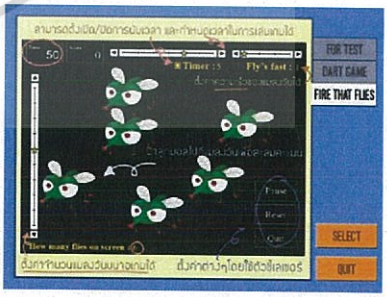
หน้าสำหรับเลือกเกมที่จะเล่น โดยใช้ Laser pointer ชี้เลือกชื่อเกมตรงแถบด้านขวาของจอภาพ ระบบจะแสดงภาพประกอบของเกมนั้น จากนั้นชี้เลือกที่ปุ่ม Select เพื่อเข้าไปเล่นเกมนั้นๆ

1) **แถบสำหรับเลือกเกม :** แถบด้านขวาบนสำหรับเลือกเกม
2) **ช่องแสดงผลภาพ :** แสดงรายละเอียดเกมที่เลือกไว้และวิธีการเล่น
3) **ตั้งค่าการเปิดปิดเสียง :** ตั้งค่าการเปิดปิดเสียงเพลงและเสียงกดปุ่ม
4) **Select :** เข้าไปยังเกมที่ได้เลือกไว้
5) **Quit :** ออกจากระบบ



ก)

ข)



ค)

รูป 3.7 หน้าเลือกเกมเมื่อเลือกเกมต่างๆ จากแถบเกมเบื้องต้น

ก) เกมพื้นฐาน
ข) เกมปาเป้า
ค) เกมปาแมลงวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.8 การตั้งค่าภายในเกม

สำหรับการตั้งค่าต่างๆ ของเกม สามารถตั้งค่าได้ดังนี้

1) **Controller Settings** : ตั้งค่าเปิด/ปิด อุปกรณ์ควบคุมการเล่นเกม
   a) **Ball** : เปิด/ปิด การใช้งานลูกบอล
   b) **Laser Pointer** : เปิด/ปิด การใช้งาน Laser Pointer
2) **Sound Setting** : เปิด/ปิด เสียงของเกม
   a) **Music** : เปิด/ปิด เสียงเพลงภายในเกม
   b) **Effect** : เปิด/ปิด เสียงของเกม
3) **Back** : ย้อนกลับไปยังหน้าเกม

3.7 การพัฒนาส่วนตรวจจับ

3.7.1 การพัฒนาส่วนตรวจจับบริเวณจอเกมที่ถูกฉายบนผนัง

เนื่องจากจอที่ถูกฉายบนผนังมีลักษณะเป็นรูปทรงสี่เหลี่ยมจึงสามารถใช้วิธีการหาวัตถุที่สนใจจากรูปทรงสี่เหลี่ยมได้ และเพื่อลดความผิดพลาดจะใช้สีในการช่วยตรวจจับ โดยขณะที่ทำการตรวจจับนั้นจะให้เครื่องฉายฉายสีที่กำหนดไว้ไปยังกำแพง

3.7.2 การพัฒนาส่วนตรวจจับลูกบอล

การตรวจจับลูกบอลจะต้องใช้ทั้งการหาวัตถุที่สนใจจากรูปทรงและสี เนื่องจากการหาด้วยรูปทรงเพียงอย่างเดียวนั้นจะทำให้เกิดข้อผิดพลาดได้ ซึ่งอาจจะเกิดได้จากการที่ในเกมมีการแสดงวัตถุที่เป็นรูปทรงกลมออกมาบนจอที่ถูกฉายบนกำแพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.3 การพัฒนาส่วนตรวจจับการกระทบผนังของลูกบอล

เนื่องจากตำแหน่งการวางกล้องนั้นหันเข้าหาผนังจึงสามารถตรวจจับการกระทบผนังของลูกบอลได้จากขนาดของลูกบอลว่าอยู่ในระยะที่ใกล้ผนังแล้วหรือไม่

3.7.4 การพัฒนาส่วนตรวจจับ Laser Pointer

การตรวจจับ Laser Pointer นั้นจะคล้ายกับการตรวจจับลูกบอลเนื่องจากเวลาชี้ Laser Pointer ไปยังกำแพงนั้นภาพที่ได้จากกล้องจะมีลักษณะเป็นรูปวงกลม แต่มีขนาดเล็กและมีความสว่างมากกว่าส่วนอื่น ๆ

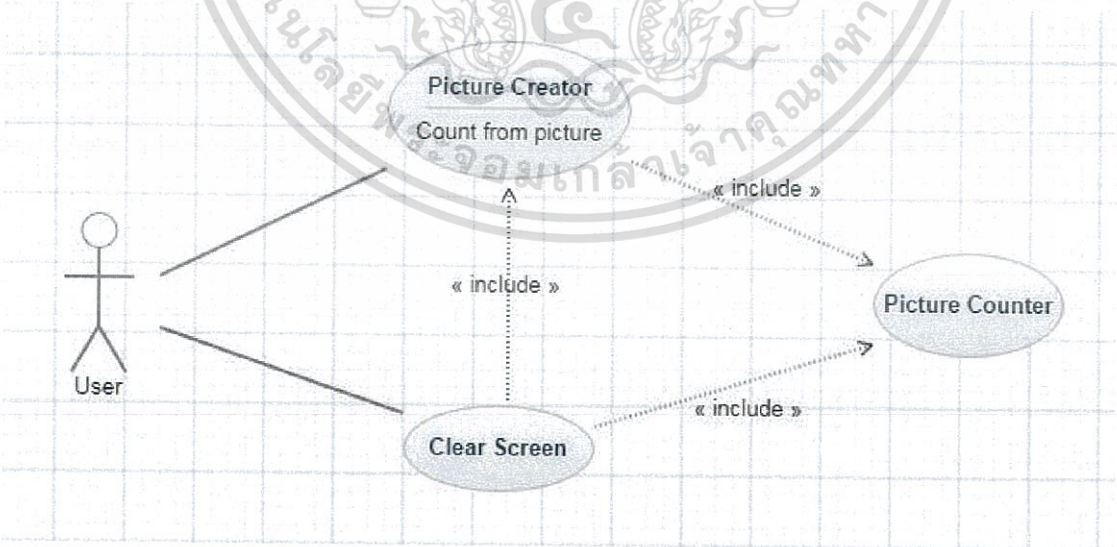
3.8 การพัฒนาตัวเกมและ Use Case Diagram

3.8.1 การพัฒนาเกมเบื้องต้น

3.8.1.1 เกมที่ 1 “ เกมพื้นฐาน ”

เกมพื้นฐาน มีส่วนประกอบต่างๆ ดังนี้

1) **Picture Creator** (ส่วนสร้างรูปการ์ตูน) สร้างรูปการ์ตูนขึ้นเมื่อลูกบอลกระทบกับจอเกม
2) **Picture Counter** (ส่วนนับจำนวนรูปการ์ตูน) นับจำนวนรูปการ์ตูนที่ปรากฏขึ้นจาก Picture Creator
3) **Clear Screen** (ส่วนล้างหน้าจอ) ลบรูปการ์ตูนที่อยู่บนจอ และล้างข้อมูลจำนวนรูปการ์ตูนใน Picture Counter



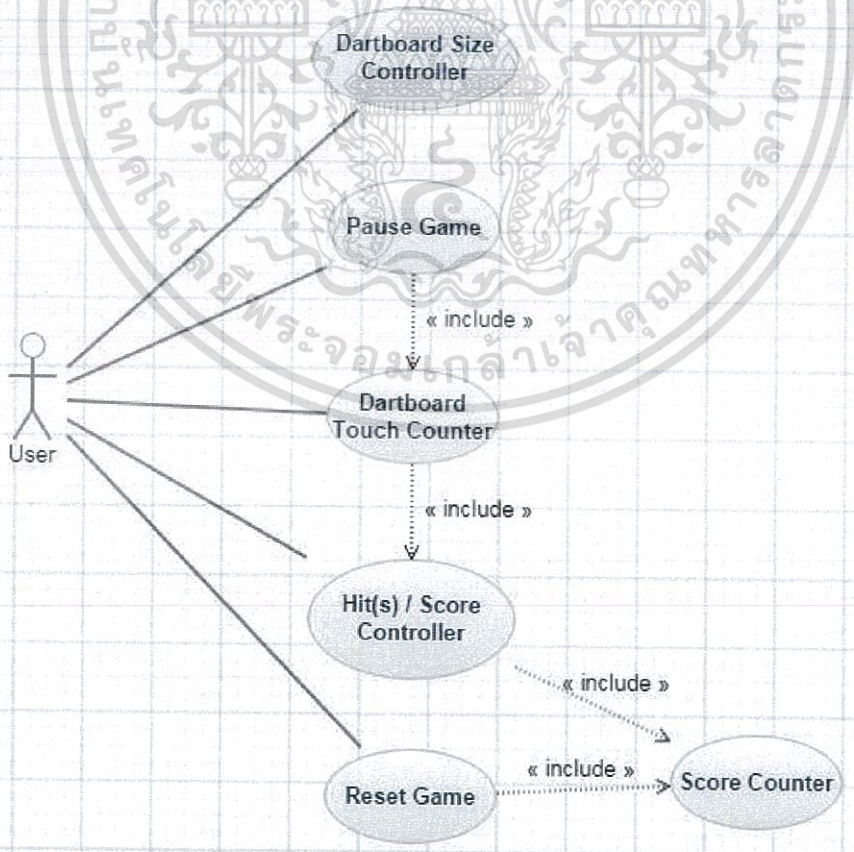
รูป 3.9 Use Case Diagram ของเกมพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.1.2 เกมที่ 2 " เกมปาเป้า "

เกมปาเป้า มีส่วนประกอบต่างๆ ดังนี้

1) **Dartboard Size Controller** (ส่วนควบคุมขนาดของเป้า) ควบคุมขนาดของเป้าตามที่ผู้ใช้กำหนด มีทั้งหมด 10 ระดับ
2) **Hit(s) / Score Controller** (ส่วนควบคุมคะแนนจากจำนวนครั้งที่ปาโดน) ควบคุมว่าจะได้ 1 คะแนน เมื่อปาลูกบอลโดนเป้ากี่ครั้ง ตามที่ผู้ใช้กำหนด มีทั้งหมด 12 ระดับ
3) **Dartboard Touch Counter** (ส่วนนับจำนวนครั้งที่ปาโดนเป้า) นับว่าผู้ใช้ปาลูกบอลโดนเป้ากี่ครั้งแล้ว
4) **Score Counter** (ส่วนนับคะแนน) คำนวณคะแนนที่ได้ โดยใช้ข้อมูลจาก Dartboard Touch Counter และ Hit(s) / Score Controller
5) **Reset Game** (ส่วน Reset เกม) ล้างข้อมูล Score Counter
6) **Pause Game** (ส่วน Pause เกม) หยุดการนับของ Dartboard Touch Counter ในขณะที่ Pause เกม



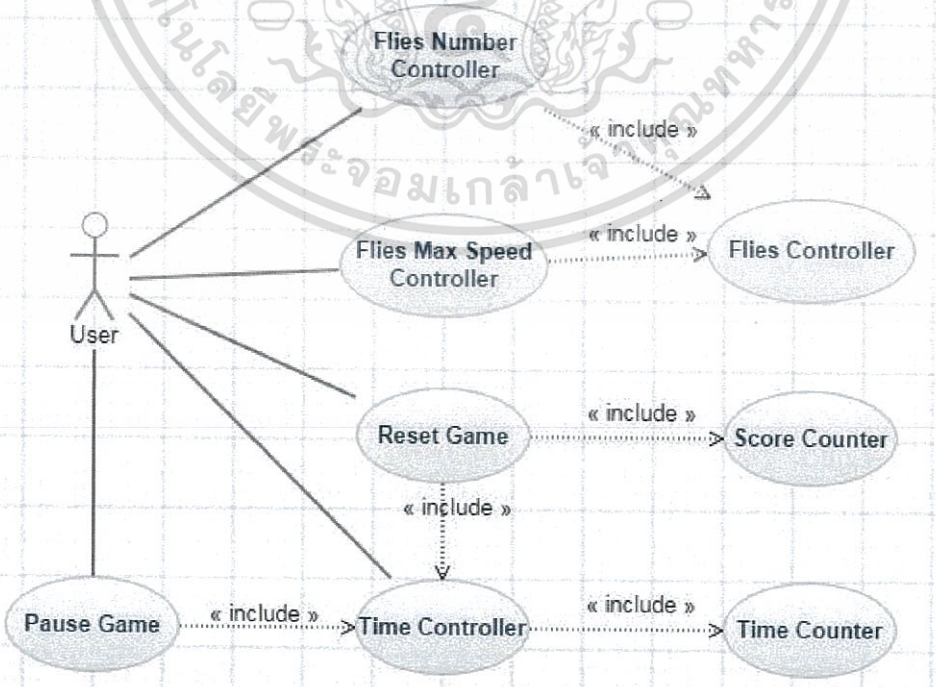
รูป 3.10 Use Case Diagram ของเกมปาเป้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.1.3 เกมที่ 3 "เกมปาแมลงวัน "

เกมปาแมลงวัน มีส่วนประกอบต่างๆ ดังนี้

1) **Flies Controller** (ส่วนควบคุมแมลงวัน) ควบคุมการเคลื่อนที่ของแมลงวันบนจอเกม และปรับค่าต่างๆของแมลงวัน โดยรับข้อมูลมาจาก Flies Number Controller และ Flies Max Speed Controller
2) **Flies Number Controller** (ส่วนควบคุมจำนวนแมลงวันบนจอเกม) ควบคุมจำนวนแมลงวันที่อยู่บนจอเกม
3) **Flies Max Speed Controller** (ส่วนควบคุมความเร็วมากที่สุดของแมลงวัน) ควบคุมความเร็วสูงสุดของแมลงวันภายในจอเกม
4) **Time Controller** (ส่วนควบคุมการนับเวลา) เปิด/ปิด และควบคุมการนับเวลาถอยหลังของเกม มีทั้งหมด 7 ระดับ ห่างกันระดับละ 10 วินาที
5) **Time Counter** (ส่วนนับเวลา) นับเวลาถอยหลังตามเวลาจริงเป็นวินาที
6) **Score Counter** (ส่วนนับคะแนน) นับคะแนนจากการที่ผู้ใช้ปาลูกบอลโดนแมลงวันบนจอเกม
7) **Pause Game** (ส่วน Pause เกม) หยุดการนับเวลาของ Time Controller
8) **Reset Game** (ส่วน Reset เกม) ล้างการนับคะแนนของ Score Controller และล้างการนับเวลาของ Time Controller



รูป 3.11 Use Case Diagram ของเกมปาแมลงวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.2 การพัฒนาหน้าเลือกเกม

การแสดงผลหน้าจอของส่วนที่ใช้ควบคุมเกมสามารถใช้ Pyglet Library ในการสร้างได้ เนื่องจาก Pyglet Library สามารถนำมาใช้สร้างหน้า Graphic User Interface และสามารถกำหนดคำสั่งให้กับการเลือกตัวเลือกต่างๆ ที่ปรากฏขึ้นมาเองได้

3.9 การพัฒนาการทำงานร่วมกันของส่วนตรวจจับและตัวเกม

การที่จะให้ส่วนตรวจจับและตัวเกมสามารถทำงานไปพร้อมๆ กันได้จะต้องใช้การทำงานแบบ Multi-Thread โดยจะให้ตัวเกมเป็น Thread หลัก ส่วนตรวจจับลูกบอลและส่วนตรวจจับ Laser เป็น Thread ย่อย เพื่อให้ส่วนต่างๆ ทำงานได้อย่างเต็มประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

เนื้อหาของบทนี้จะเกี่ยวกับการทดสอบการทำงานของหน้าเลือกเกม การรับภาพจากกล้อง ประมวลผล แล้วแสดงผลออกมาบนจอภาพ โดยจะทำการทดลองการตรวจจับลูกบอลและการกระทบผนังของลูกบอล ตรวจจับ Laser Pointer การทดลองแบ่ง Thread และการทดลองพัฒนาตัวเกมด้วย Pyglet Library เพื่อให้เห็นผลลัพธ์ของการพัฒนาชิ้นงานในแต่ละส่วน ก่อนจะนำมารวมกันเป็นชิ้นงาน " ON THE BALL " GAMES PLAYER

4.1 การทดลองการตรวจจับลูกบอล

4.1.1 การทดลองเกี่ยวกับ

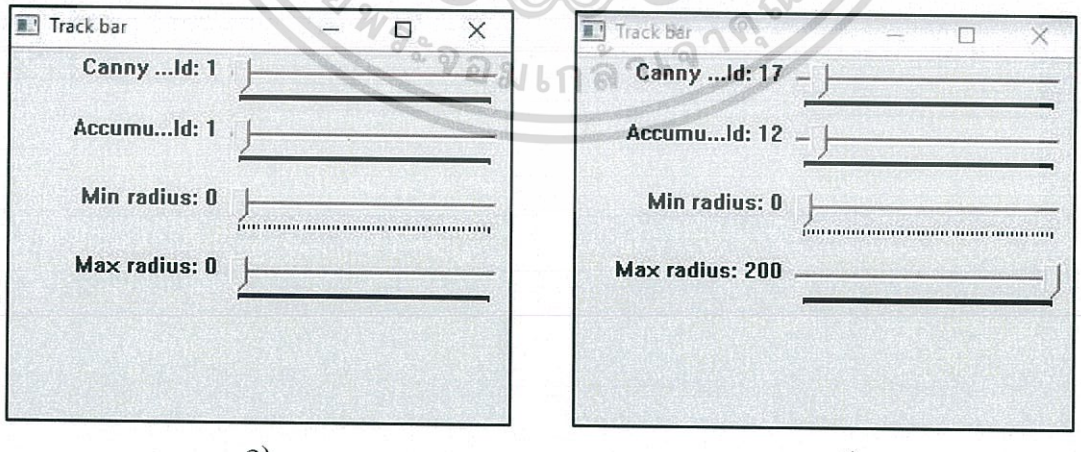
ค่าของตัวแปรต่างๆ ที่มีผลต่อการตรวจจับลูกบอล

4.1.2 วัตถุประสงค์ของการทดลอง

เพื่อหาค่าที่เหมาะสมของตัวแปรต่างๆ ที่จะทำให้สามารถตรวจจับลูกบอลได้อย่างถูกต้องแม่นยำ

4.1.3 ทดลองการตรวจจับลูกบอลด้วยรูปร่าง

การตรวจจับลูกบอลด้วยรูปร่างใน OpenCV นั้นใช้ฟังก์ชั่น HoughCircles คือฟังก์ชั่นที่ใช้หาวงกลมในรูปเฉดสีเทาโดยใช้ Hough transform ทดลองโดยการปรับค่าต่างๆ ของ Hough Transform ในฟังก์ชั่น เพื่อให้โปรแกรมสามารถตรวจจับวงกลมได้แม่นยำขึ้น



ก)

ข)

รูป 4.1 Track Bar สำหรับปรับค่าต่างๆ ของ Hough Transform ในฟังก์ชั่น

ก) Track Bar

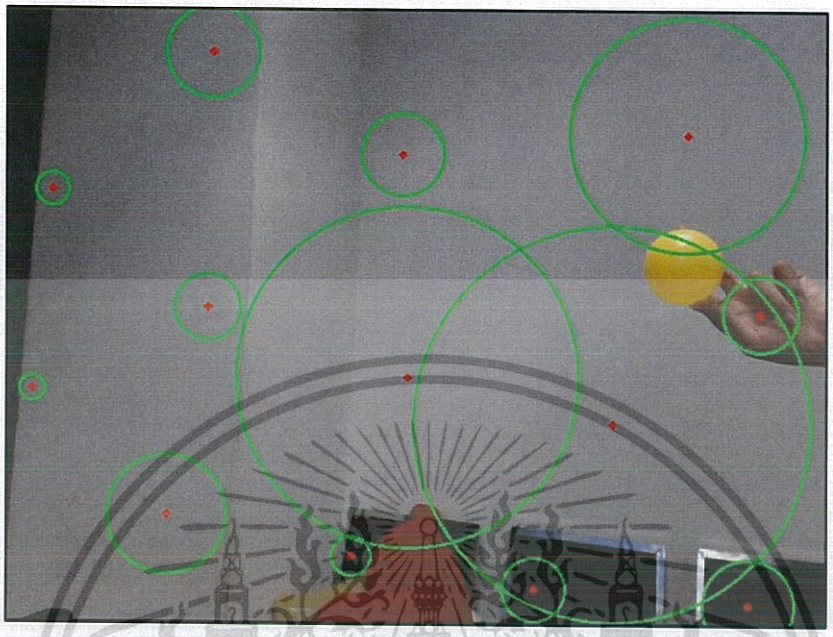
ข) Track Bar ที่ถูกเลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยค่าที่สำคัญในการตรวจจับลูกบอลนั้นมีอยู่ 2 ค่า คือ

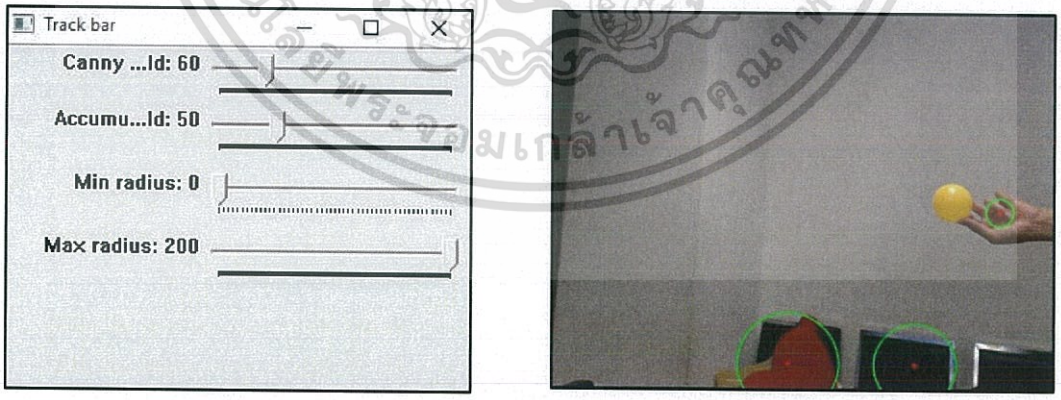
1) Canny Threshold (แถวที่ 1 ของรูป 4.1)

2) Accumulator Threshold (แถวที่ 2 ของรูป 4.1)



รูป 4.2 ผลลัพธ์การตรวจจับ

เริ่มต้นเมื่อค่า Canny Threshold และ Accumulator Threshold น้อย จะทำให้ไม่สามารถตรวจจับลูกบอลได้ดังรูป 4.1 ก) และรูป 4.2 ดังนั้นจึงต้องเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น



ก) ข)

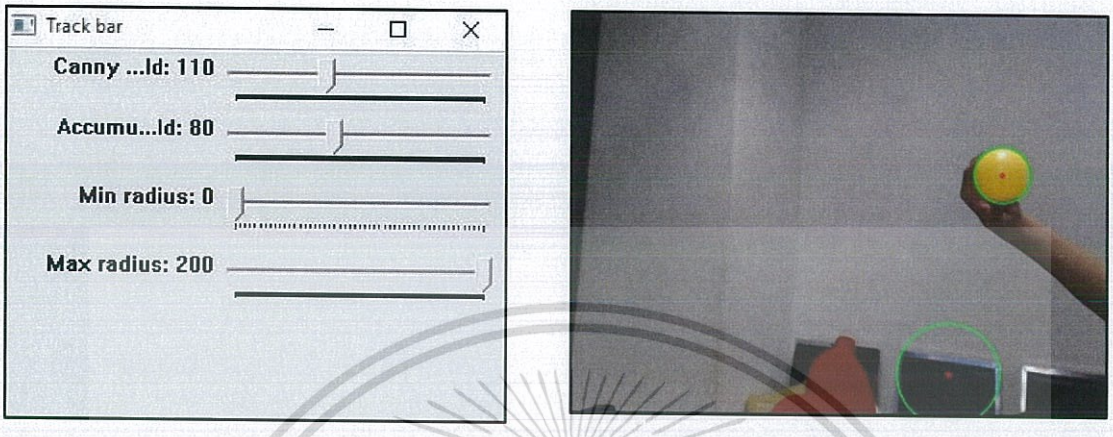
รูป 4.3 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น

ก) Track Bar ที่ถูกเลื่อนมากขึ้น

ข) ผลลัพธ์การตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.3 ก) มีการเพิ่มค่า Canny Threshold และ Accumulator Threshold มากขึ้น จึงลดความผิดพลาดในการตรวจจับวงกลม แต่ยังไม่สามารถตรวจจับวงกลมที่ต้องการได้ ดังผลการทดลองรูป 4.3 ข) จึงต้องเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้นอีก



ก) ข)

รูป 4.4 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold มากขึ้นอีก

ก) Track Bar ที่ถูกเลื่อนมากขึ้น

ข) ผลลัพธ์การตรวจจับ

จากรูป 4.4 ข) จะเห็นได้ว่าการตรวจจับวงกลมมีความแม่นยำมากขึ้น แต่ยังไม่มีประสิทธิภาพมากพอที่จะนำไปใช้งาน จึงต้องทำการเบลอรูปที่นำมาใช้ในการตรวจจับโดยใช้ "GaussianBlur" เพื่อลด Noise ที่เกิดขึ้นในภาพที่นำมาทำการตรวจจับวงกลม

ก) ข)

รูป 4.5 ใช้ GaussianBlur เพื่อลด Noise ที่เกิดขึ้นในภาพที่นำมาตรวจจับ

ก) หลังการทำ GaussianBlur

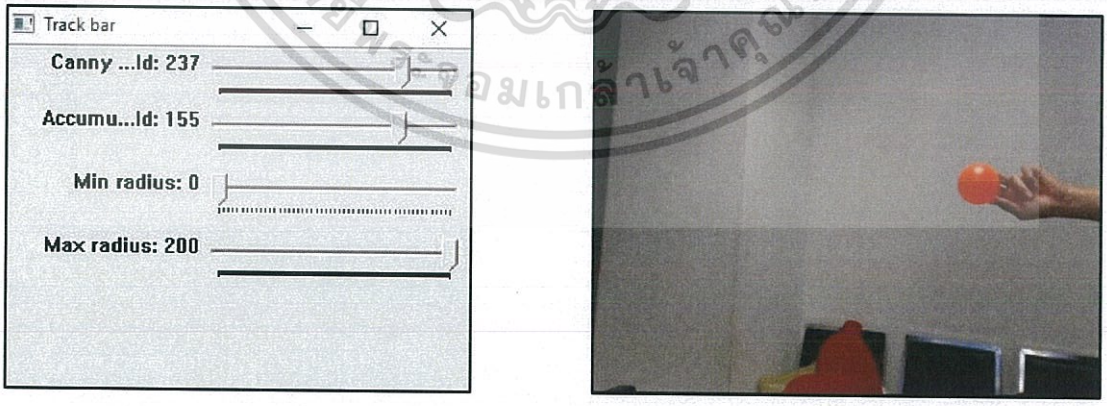
ข) ผลลัพธ์การตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากใช้ " GaussianBlur " ทำการเบลอภาพ การตรวจจับที่ผิดพลาดจะลดน้อยลงโดยที่ไม่ต้องเพิ่มค่า Canny Threshold และ Accumulator Threshold แต่หากในภาพมีเส้นโค้งเกิดขึ้นและค่า Canny Threshold กับ Accumulator Threshold ยังไม่มากพอ ก็ทำให้เกิดข้อผิดพลาดได้เช่นกัน

รูป 4.6 ตรวจเจอเส้นโค้งที่ไม่ใช่วงกลม

จากรูป 4.6 มีเส้นโค้งอยู่ในภาพแต่ฟังก์ชั่น HoughCircle เห็นเป็นวงกลม แสดงว่าค่า Canny Threshold และ Accumulator Threshold ยังน้อยเกินไป แต่ถ้ากำหนดค่าให้ Canny Threshold และ Accumulator Threshold มากเกินไป ทำให้ตรวจจับไม่พบวงกลมหรือลูกบอลที่ต้องการได้



ก)

ข)

รูป 4.7 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น

ก) Track Bar ที่ถูกเลื่อน

ข) ผลลัพธ์การตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นต้องทำการลดค่า Canny Threshold และ Accumulator Threshold ให้พอดีจนกว่าจะตรวจจับเจอลูกบอล โดยที่เกิดความผิดพลาดน้อยที่สุด รวมถึงทำการปรับค่า Minimum Radius และ Maximum Radius ให้เท่ากับขนาดใหญ่ที่สุดและเล็กที่สุดของลูกบอลที่เห็นในภาพตามลำดับ เพื่อลดข้อผิดพลาด

จากการทดลองที่ผ่านมาจะเห็นได้ว่าการตรวจจับนั้นยังไม่มีประสิทธิภาพมากพอที่จะตรวจจับลูกบอลที่ใช้ในการเล่นเกม เนื่องจากยังตรวจจับได้ไม่ตลอดเวลา ดังนั้นจึงต้องใช้สีช่วยในการตรวจจับลูกบอล

4.1.4 ทดลองการตรวจจับลูกบอลด้วยสี

เนื่องจากการตรวจจับลูกบอลด้วยรูปร่างเพียงอย่างเดียวนั้นยังมีความแม่นยำไม่เพียงพอ ดังนั้นต้องใช้การตรวจจับสีของลูกบอลเพื่อช่วยในการระบุตำแหน่งของลูกบอล โดยการเลือกพื้นที่เฉพาะสีที่สนใจในภาพมาก่อน แล้วค่อยดูว่าพื้นที่ของสีที่สนใจนั้นเป็นรูปวงกลมหรือไม่โดยการใช้คำสั่ง inRange ของ OpenCV ในการเลือกสีที่สนใจตามสีของลูกบอล โดยค่าสีที่นำมาใช้เลือกจะเป็นแบบ HSV

การเลือกสีของลูกบอลทำได้โดยการนำลูกบอลที่ต้องการเข้ามาในจอ แล้วค่อยๆเพิ่มหรือลดค่าของช่วงสีแต่ละค่าจนกว่าจะได้ผลลัพธ์เป็นพื้นที่วงกลมของลูกบอลที่นำมาหาค่าสี โดยจะต้องทำการเลือกในสภาพแวดล้อมที่ใช้งานจริงเนื่องจากแสงและเงาแต่ละสถานที่ไม่เหมือนกัน

ก) ข) ค)

รูป 4.8 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น

ก) Track Bar HSV

ข) วิดีโอที่มีลูกบอลอยู่

ค) พื้นที่สี

ในรูป 4.8 ก) จะยังไม่มีการเปลี่ยนช่วงของค่าสีดังนั้นในรูป 4.8 ค) จึงได้ผลลัพธ์เป็นสีขาวทั้งรูป (สีขาวคือพื้นที่ที่สนใจ) หลังจากนั้นจะทำการเลื่อนค่ามากสุดและน้อยสุดของแต่ละค่าเข้าหากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก) ข) ค)

รูป 4.9 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น

ก) Track Bar HSV

ข) วิดีโอที่มีลูกบอลอยู่

ค) พื้นที่สี

ในรูป 4.9 ก) ได้ทำการเลื่อนช่วงของค่าต่างๆ เข้าหากันแล้ว จึงเกิดสีดำขึ้นในรูป 4.9 ค) ซึ่งเป็นพื้นของค่าสีที่ได้ถูกตัดออกไป จากนั้นเลื่อนช่วงของค่าสีต่างๆ เข้าหากันมากขึ้นจนกว่าจะเห็นรูปวงกลมสีขาว

ก) ข) ค)

รูป 4.10 เมื่อเพิ่มค่า Canny Threshold และ Accumulator Threshold ให้มากขึ้น

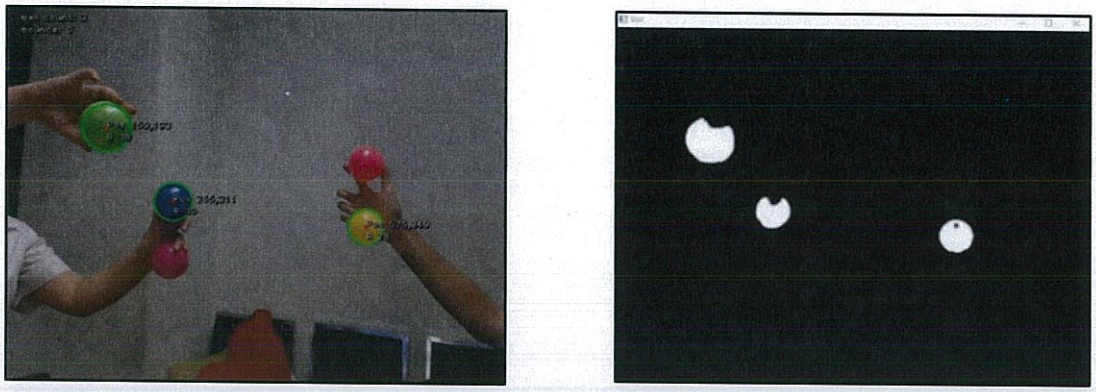
ก) Track Bar HSV

ข) วิดีโอที่มีลูกบอลอยู่

ค) พื้นที่ของลูกบอลที่ได้

ในรูป 4.10 ค) จะเห็นพื้นที่สีขาวเป็นรูปวงกลมชัดเจน จึงนำค่าที่ได้จากรูป 4.10 ก) ไปใช้ในการกำหนดค่าสีของลูกบอลที่นำมาทดสอบ ทำจนครบตามสีของลูกบอลที่นำมาใช้แล้วนำพื้นที่ที่ได้มาเข้าฟังก์ชั่น HoughCircles เพื่อหาตำแหน่งและขนาดของลูกบอลในภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ข)

รูป 4.11 เมื่อนำพื้นที่ที่ได้มาเข้าฟังก์ชั่น HoughCircles

ก) การตรวจจับบนวิดีโอที่มีลูกบอลอยู่

ข) พื้นที่ของลูกบอลสีที่ได้

จากรูป 4.11 ก) จะเห็นว่าการตรวจจับลูกบอลมีความแม่นยำเพิ่มขึ้นเป็นอย่างมากจนสามารถตรวจจับลูกบอลได้ครบ และไม่ตรวจจับลูกบอลสีที่ไม่ได้ถูกกำหนดไว้

ตาราง 4.1 ค่าสี HSV ของบอลแต่ละสี

	H_MIN	H_MAX	S_MIN	S_MAX	V_MIN	V_MAX
Yellow ball	20	25	110	235	154	256
Red ball	0	180	200	256	120	210
Green ball	40	60	75	185	77	222
Orange ball	2	9	165	256	180	256
Blue ball	105	115	160	256	90	256

4.1.5 ผลการทดลอง

ค่าตัวแปรที่เหมาะสมทำให้ส่วนตรวจจับสามารถตรวจจับลูกบอลได้อย่างแม่นยำ

4.2 การทดลองตรวจจับการกระทบผนังของลูกบอล

4.2.1 การทดลองเกี่ยวกับ

การตรวจจับการกระทบผนังของลูกบอล

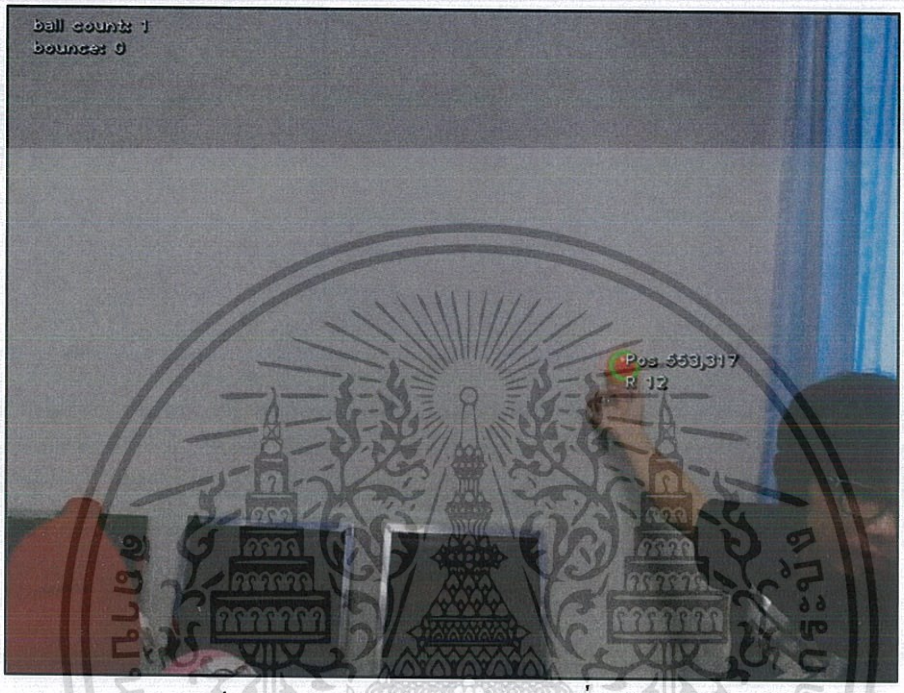
4.2.2 จุดประสงค์ของการทดลอง

เพื่อพิสูจน์ว่าวิธีที่ใช้การตรวจจับขนาดของลูกบอลเมื่ออยู่ในระยะกระทบกำแพง ใช้ตรวจจับการกระทบผนังของลูกบอลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ทดลองการตรวจจับการกระทบผนังด้วยขนาดของลูกบอล

เราสามารถใช้ขนาดประมาณของลูกบอลในเวลาที่กระทบผนังมาใช้ในการทดสอบการตอบสนองเบื้องต้นของเกมก่อนได้ โดยการตั้งกล้องที่ใช้ในการตรวจจับลูกบอลให้หันตรงเข้าหาผนัง แล้วนำลูกบอลไปติดไว้ที่ผนัง เพื่อหาขนาดลูกบอลในขณะที่กระทบกับผนัง



รูป 4.12 ลูกบอลถูกตรวจจับขณะที่อยู่ติดผนัง

จากนั้นลองใช้ช่วงของขนาดต่างๆ ที่ลูกบอลจะชนผนัง โดยจะให้มีขนาดใหญ่กว่าขนาดที่หาไว้ก่อนหน้านี้เล็กน้อย แล้วทดลองโยนลูกบอลใส่ผนัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.13 รูปการ์ตูนขึ้นมาบนจอภาพ หลังจากลูกบอลกระทบผนัง

เนื่องจากมีลูกบอลหลายลูก จึงต้องทำการนับลูกบอลที่เข้ามาใหม่เพื่อลดโอกาสที่จะเกิดภาพซ้ำหรือผิดตำแหน่ง

```
C:\Python27\python.exe C:/Users/Guide-ce/Documents/Project/Example/ColorTrack/ColorTracking.py
increase
ball before = 0
ball count = 1
increase
ball before = 1
ball count = 2
increase
ball before = 1
ball count = 2
increase
ball before = 2
ball count = 3
increase
ball before = 2
ball count = 3
increase
ball before = 0
ball count = 1
```

รูป 4.14 การนับจำนวนบอลที่เข้ามาใหม่ใน Console

การตรวจจับการกระทบผนังด้วยวิธีดูจากขนาดของลูกบอลเพียงอย่างเดียว ยังคงมีข้อผิดพลาดเกิดขึ้นอยู่บ้าง แต่ยังพอใช้ได้สำหรับการทดลองการตอบสนองเบื้องต้นของเกม

4.2.4 ผลการทดลอง

สามารถตรวจจับการกระทบกำแพงของลูกบอลโดยใช้ขนาดลูกบอลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองปรับปรุงการตรวจจับการกระทบผนังโดยใช้ทิศทางการเคลื่อนที่ของลูกบอล

4.3.1 การทดลองเกี่ยวกับ

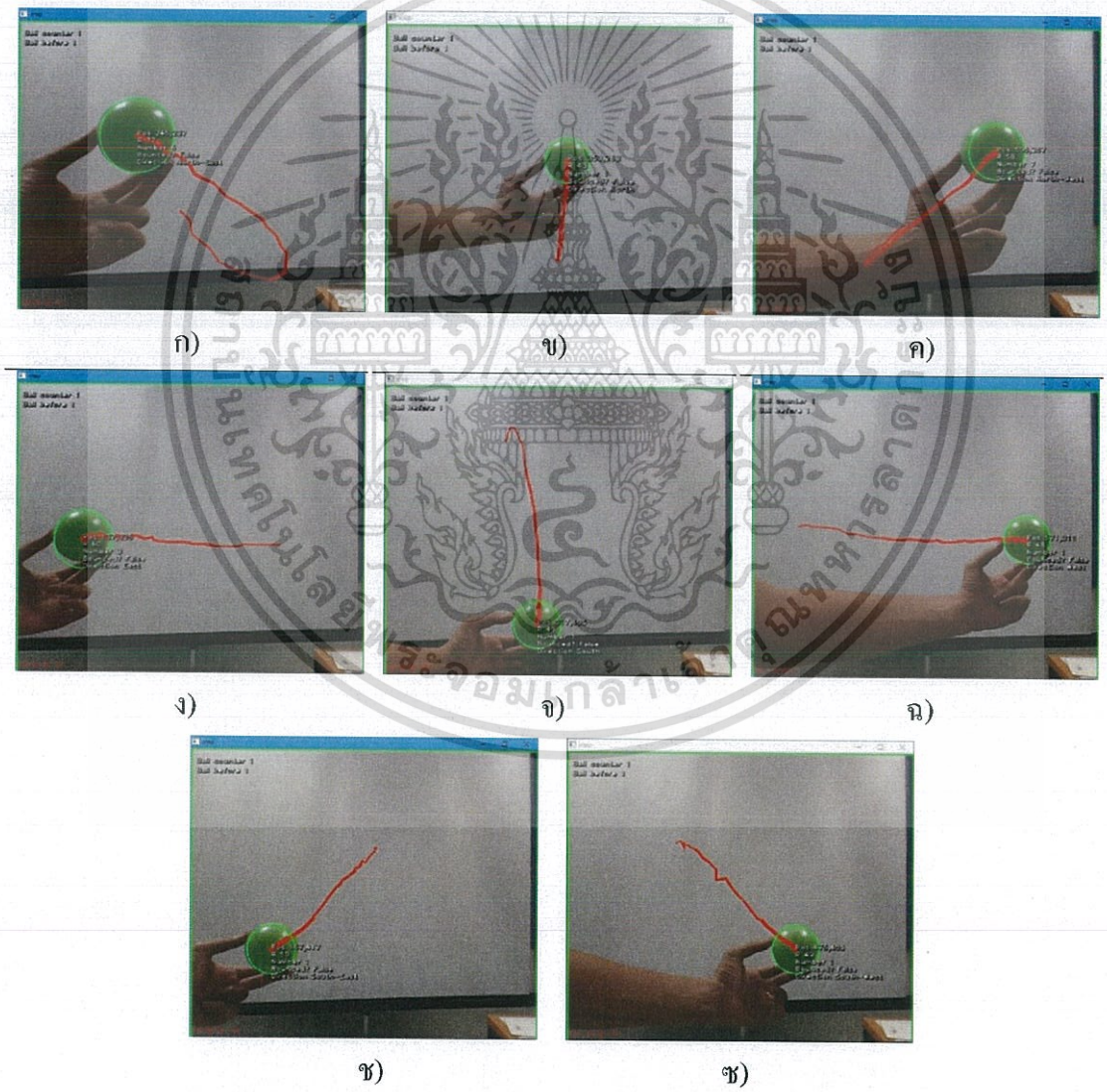
การปรับปรุงการตรวจจับการกระทบผนังของลูกบอลโดยใช้ทิศทางการเคลื่อนที่ของลูกบอล

4.3.2 จุดประสงค์ของการทดลอง

เพื่อทดสอบการนำทิศทางการเคลื่อนที่ของลูกบอลมาใช้ในการตรวจจับการกระทบผนังของลูกบอล ว่าสามารถนำมาใช้ได้จริงหรือไม่

4.3.3 ทดลองการตรวจจับการกระทบผนังด้วยขนาดของลูกบอล

หาทิศทางการเคลื่อนที่ของลูกบอลทั้ง 8 ทิศ โดยใช้การเปรียบเทียบพิกัดตำแหน่งของลูกบอลในเวลาต่างๆ ตามรูป 4.15



รูป 4.15 การตรวจจับการเคลื่อนที่ของลูกบอลในทิศทางต่างๆ

ก) ทิศตะวันตกเฉียงเหนือ

ข) ทิศเหนือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค) ทิศตะวันออกเฉียงเหนือ

ง) ทิศตะวันตก

จ) ทิศใต้

ฉ) ทิศตะวันออก

ช) ทิศตะวันตกเฉียงใต้

ซ) ทิศตะวันออกเฉียงใต้

4.3.4 ผลการทดลอง

ไม่สามารถตรวจจับการกระทบผนังของลูกบอลโดยใช้การตรวจจับทิศทางการเคลื่อนที่ได้ เนื่องจากระบบที่ได้วางไว้มีกล้องเพียงตัวเดียว ซึ่งวางอยู่ในตำแหน่งด้านหน้าของจอ ทำให้กล้องเห็นการเปลี่ยนแปลงทิศทางการเคลื่อนที่ของลูกบอลก่อนและหลังการกระทบผนังแทบไม่แตกต่างกัน เช่น ในกรณีที่ลูกบอลเข้ามายังหน้าจอจากด้านบน ลูกบอลจะมีทิศทางก่อนการกระทบผนังเป็นเคลื่อนที่ไปทางทิศใต้ และหลังจากที่ลูกบอลกระทบผนังแล้ว ลูกบอลก็จะยังคงมีทิศทางในการเคลื่อนที่ไปทางทิศใต้ ซึ่งเหมือนกับทิศทางก่อนการกระทบผนัง ทำให้ไม่สามารถตรวจจับการกระทบผนังของลูกบอลโดยใช้ทิศทางการเคลื่อนที่ของลูกบอลได้

4.4 การทดลองตรวจจับ Laser Pointer

4.4.1 การทดลองเกี่ยวกับ

การตรวจจับ Laser Pointer ซึ่งใช้เป็นตัวชี้เพื่อปรับตั้งค่าต่างๆ บนจอเกม

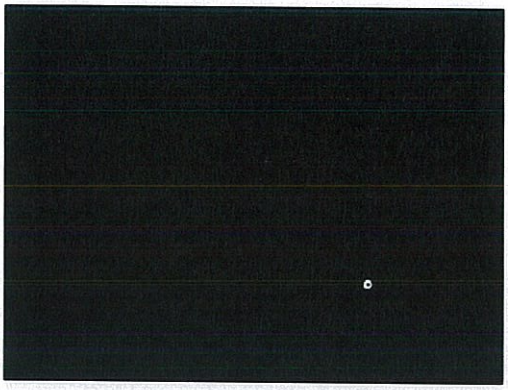
4.4.2 วัตถุประสงค์ของการทดลอง

ทดลองการตรวจจับ Laser Pointer ว่าวิธีการที่เลือกนำมาใช้ สามารถตรวจจับ Laser Pointer ได้อย่างมีประสิทธิภาพหรือไม่ อย่างไร

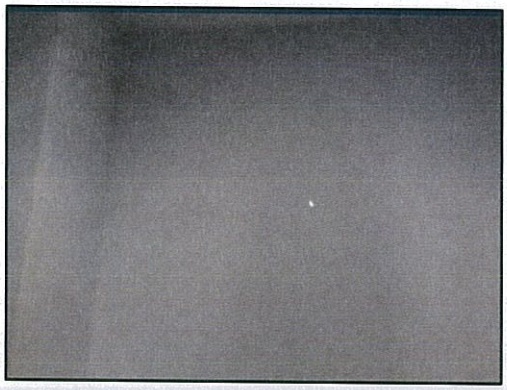
4.4.3 ขั้นตอนการทดลอง

เนื่องจาก Laser Pointer นั้นมีลักษณะเป็นทรงกลมจึงใช้ฟังก์ชั่น HoughCircles และใช้สีในการช่วยกรองหาแสงจาก Laser Pointer เหมือนกับการหาลูกบอล ดังรูป 4.16 ก) และ 4.16 ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก)



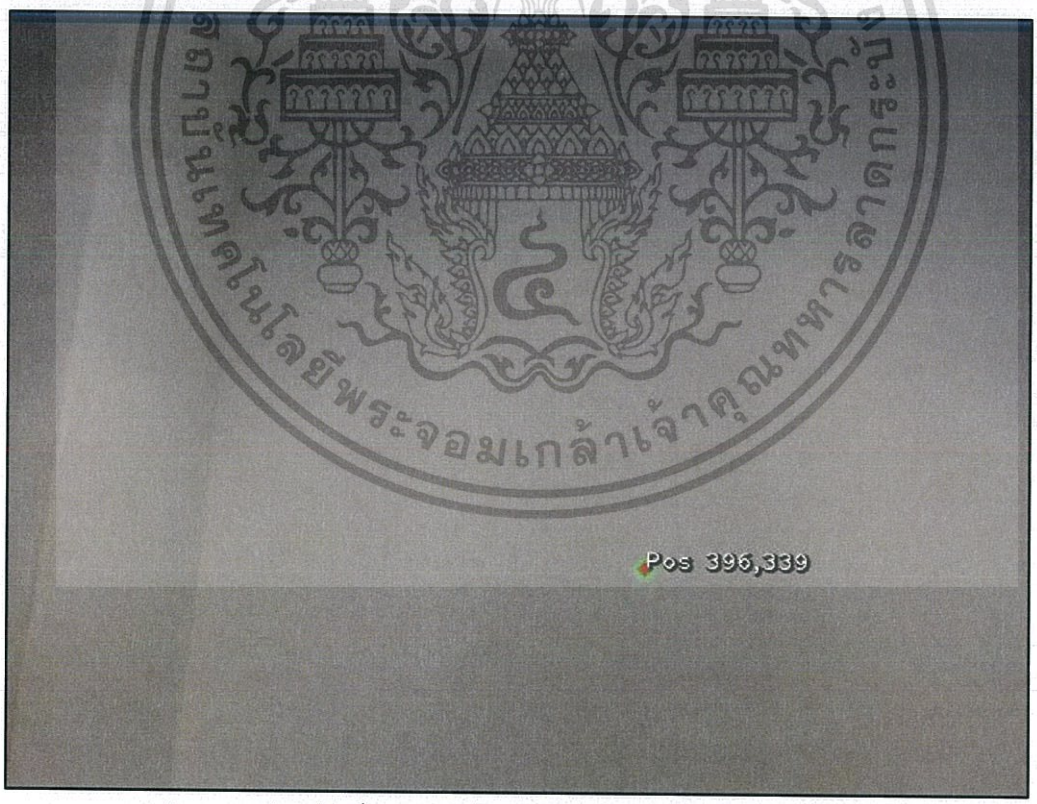
ข)

รูป 4.16 รูปการตรวจจับ Laser Pointer

ก) สามารถเลือกเฉพาะพื้นที่ของแสงจาก Laser Pointer ออกมาได้

ข) สามารถเห็นแสงจาก Laser Pointer เป็นวงกลมได้

เพื่อลดความผิดพลาดในการตรวจจับจึงกำหนดขนาดสูงสุดของวงกลมที่ได้จากฟังก์ชั่น HoughCircle ไว้ จะได้ตำแหน่งของแสงจาก Laser Pointer ดังรูป 4.17



รูป 4.17 รูปผลลัพธ์ของการปรับขนาดการตรวจจับ Laser Pointer

การตรวจจับ Laser Pointer นั้นจะได้ค่าตำแหน่งของ Laser มาหลายครั้งในหนึ่งวินาที ตามความเร็วของการตรวจจับจึงต้องทำการลดความเร็วของแต่ละรอบการตรวจจับลงเพื่อไม่ให้เกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งค่าตำแหน่งซ้ำ เพราะจะทำให้การทำงานของเกมผิดพลาดได้ และที่สามารถชะลอการทำงานของส่วนตรวจจับ Laser Pointer ได้เนื่องจาก Laser Pointer นั้นนำมาใช้เฉพาะการเลือกคำสั่งจึงไม่จำเป็นต้องมีความต่อเนื่องในการตรวจจับ โดยสามารถกำหนดความเร็วของรอบการตรวจจับได้จากการใช้ฟังก์ชั่นหน่วงเวลาเพื่อคั่นการทำงานในแต่ละรอบให้ช้าลง

4.4.4 ผลการทดลอง

วิธีการที่เลือกมาสามารถตรวจจับ Laser Pointer ได้อย่างมีประสิทธิภาพ

4.5 การทดลองแบ่ง Thread

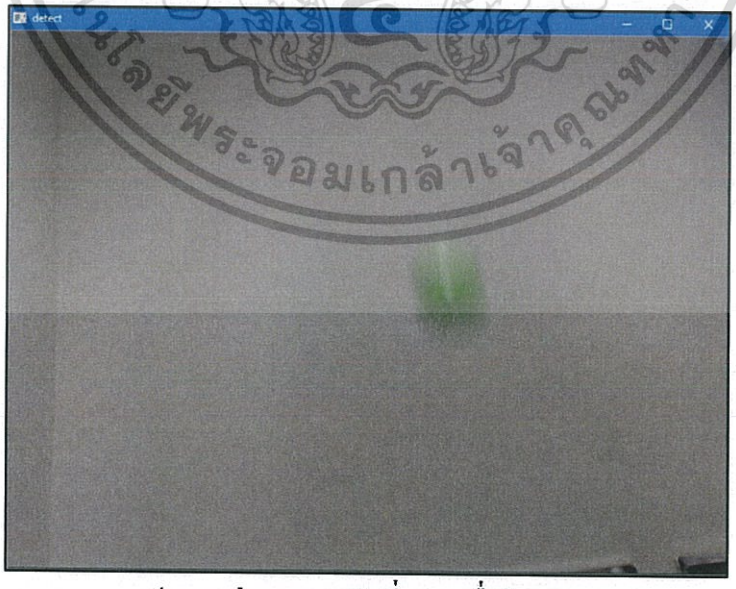
4.5.1 การทดลองเกี่ยวกับ

ทดลองใช้การแบ่ง Thread เพื่อแยกการทำงานระหว่างส่วนของส่วนตรวจจับและส่วนของเกม

4.5.2 วัตถุประสงค์ของการทดลอง

ในการทำงานของเกมและส่วนตรวจจับตอนแรกนั้นทำโดยการใช้ Clock Interval ซึ่งเป็นฟังก์ชั่นของ Pyglet Library ที่ใช้ในการเขียนเกม ซึ่งจะเรียกใช้ฟังก์ชั่นที่เรากำหนดให้ ให้ทำงานตามเวลาที่กำหนดไว้ ส่วนประมวลผลจึงต้องแบ่งกันทำงานระหว่างตัวเกมและส่วนตรวจจับ ทำให้ทั้งสองส่วนทำงานได้ช้ากว่าที่ควรจะเป็น ส่งผลให้ส่วนตรวจจับไม่สามารถตรวจจับลูกบอลที่เคลื่อนที่เร็วมากๆ ได้ ดังรูป 4.18

จึงทดลองนำ Thread มาใช้แยกการทำงานของส่วนตรวจจับและตัวเกมออกจากกัน



รูป 4.18 ส่วนตรวจจับที่เบลอเมื่อโยนบอล

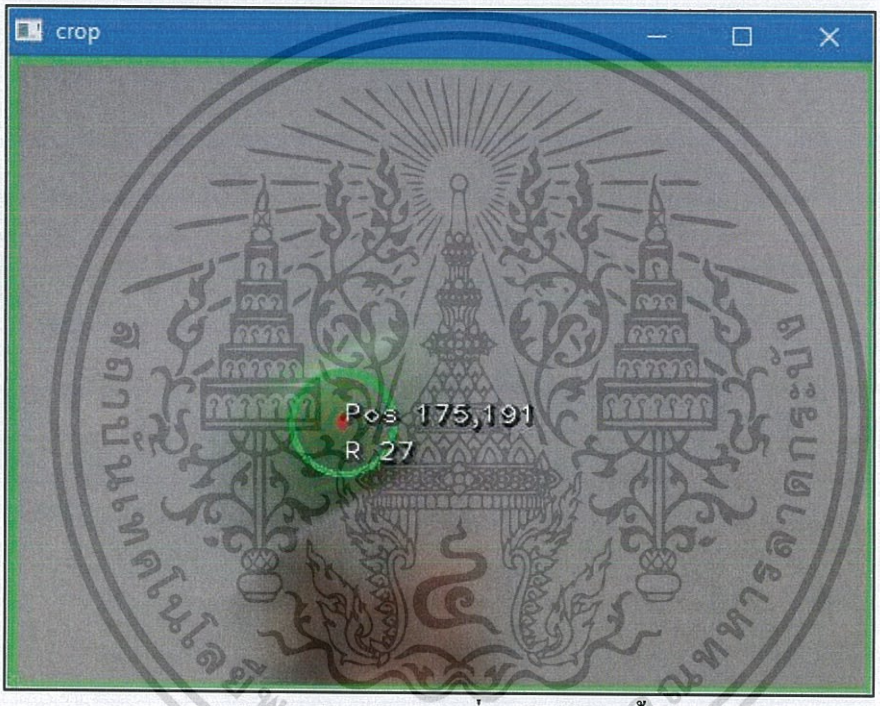
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 ขั้นตอนการทดลอง

ทดลองแก้ไขด้วยการ แยกส่วนของตัวเกมออกมาเป็น Thread หลัก ส่วนตรวจจับออกมาเป็น Thread ย่อย ซึ่งเป็นลูกของ Thread หลัก เนื่องจากการทำงานของส่วนตรวจจับนั้นเป็นลักษณะการทำงานแบบทำงานไปเรื่อยๆ ดังนั้นจะไม่สามารถออกจากเกมได้ จึงต้องตั้งให้ Thread ของส่วนตรวจจับเป็นแบบ Daemon จะทำให้สามารถจบการทำงานของเกมได้ตามปกติ

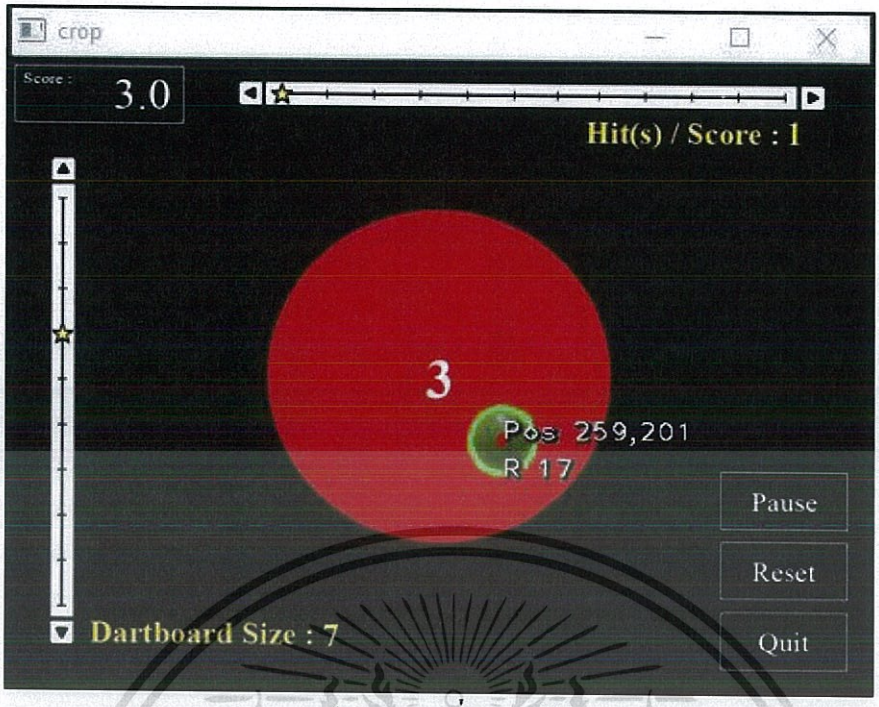
4.5.4 ผลการทดลอง

ตรวจจับลูกบอลได้มีประสิทธิภาพมากยิ่งขึ้น สามารถตรวจจับลูกบอลที่เคลื่อนที่เร็วได้แล้ว ดังรูป 4.19 ทำให้ตัวเกมทำงานได้ตามปกติดังรูป 4.20



รูป 4.19 ส่วนตรวจจับที่ทำงานได้เร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.20 เกมที่ทำงานได้

4.6 การทดลองพัฒนาเกมด้วย Pyglet Library

4.6.1 การทดลองเกี่ยวกับ

การพัฒนาตัวเกมด้วย Pyglet Library

4.6.2 วัตถุประสงค์ของการทดลอง

ทดสอบว่า Pyglet Library สามารถนำมาใช้ในการพัฒนาตัวเกมเบื้องต้นเพื่อใช้ร่วมกับส่วนตรวจจับได้หรือไม่

4.6.3 การทดลองสร้างหน้าจอเกม

4.6.3.1 ขั้นตอนการทดลอง

สร้างหน้าจอเกมด้วยการเขียนฟังก์ชั่นดังโปรแกรม 4.1 โดยในบรรทัดที่ 3 เป็นการป้อนค่าตัวเลขเพื่อกำหนดว่าจะให้หน้าจอเกมแสดงผลบนอุปกรณ์ใด โดย 0 คือหน้าจอหลักของคอมพิวเตอร์ที่ใช้ในการประมวลผลฟังก์ชั่น 1 คือหน้าจอที่ต่อเพิ่มเข้ามา เช่น จอเสริม เครื่องฉาย เป็นต้น ในบรรทัดที่ 4 เป็นการตั้งค่าหน้าจอให้แสดงผลแบบ Fullscreen และตั้งค่าขนาดความกว้าง ความสูงของหน้าจอเกม ส่วนบรรทัดต่อมาเป็นการสั่งให้โปรแกรมแสดงผลหน้าจอขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 4.1 การสร้างหน้าจอเกม

```
platform = pyglet.window.get_platform()
display = platform.get_default_display()
screens = display.get_screens()[1]
game_window = pyglet.window.Window( fullscreen=True,
    screen=screens, width=WIDTH, height=HEIGHT)

if __name__ == '__main__':
```

4.6.3.2 ผลการทดลอง

จะได้หน้าจอเกมดังรูป 4.21

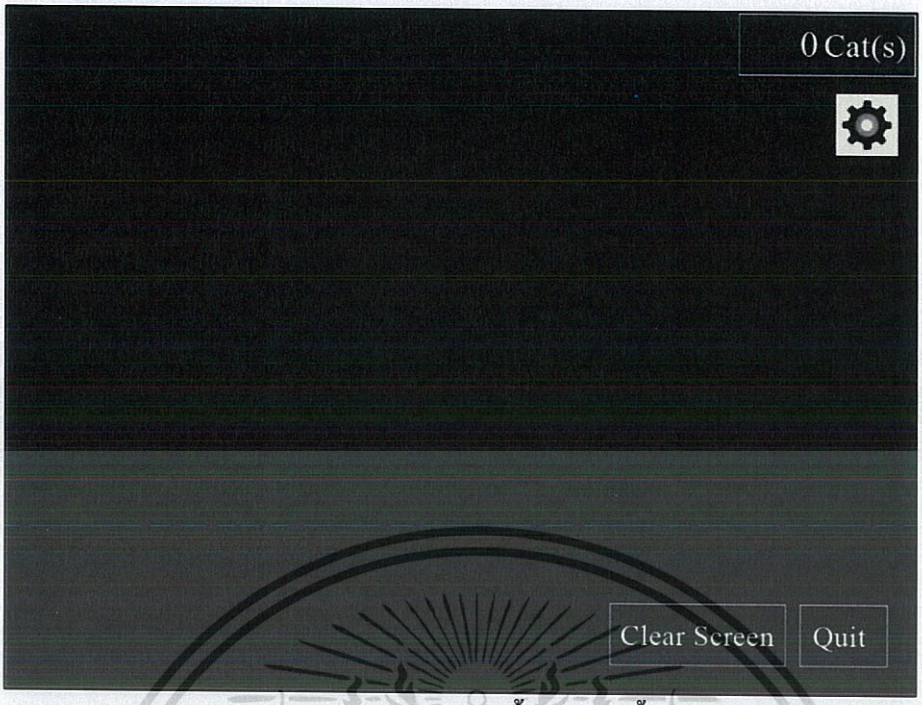
รูป 4.21 หน้าจอเกมที่ได้จากโปรแกรม 4.1

4.6.4 การทดลองใส่รายละเอียดของเกมเบื้องต้น

4.6.4.1 เกมที่ 1 " เกมพื้นฐาน "

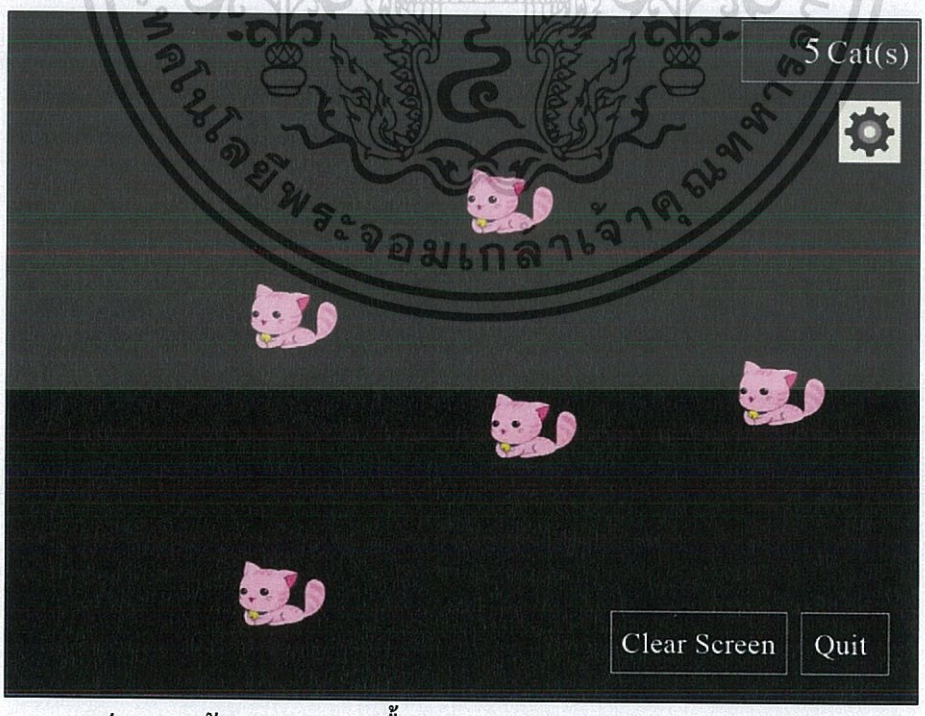
รายละเอียดเบื้องต้นของเกมที่ต้องใส่ได้แก่ช่องบอกจำนวนรูปภาพแมวที่ปรากฏบนจอจากการปาลูกบอล และปุ่มสำหรับใช้ Laser Pointer ชี้เพื่อ Reset จำนวนและรูปภาพแมวบนหน้าจอ ปุ่มสำหรับเข้าไปยังหน้าตั้งค่าเกม และปุ่มสำหรับออกจากเกม ดังรูป 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.22 หน้าจอเกม " เกมพื้นฐาน " เบื้องต้น

จากนั้น เขียนโปรแกรม เพื่อให้แสดงผลรูปภาพแมวขึ้นในตำแหน่งที่บอลกระทบจอเกมเมื่อลูกบอลกระทบจอเกม และแสดงผลจำนวนรูปภาพแมวที่ขึ้นมาที่ช่องด้านขวาบน ดังรูป 4.23 จากหน้าจอนี้ ผู้ใช้สามารถใช้ Laser Pointer ชี้ที่ปุ่ม Clear Screen เพื่อ Reset เกม



รูป 4.23 หน้าจอเกม " เกมพื้นฐาน " หลังจากลูกบอลกระทบหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

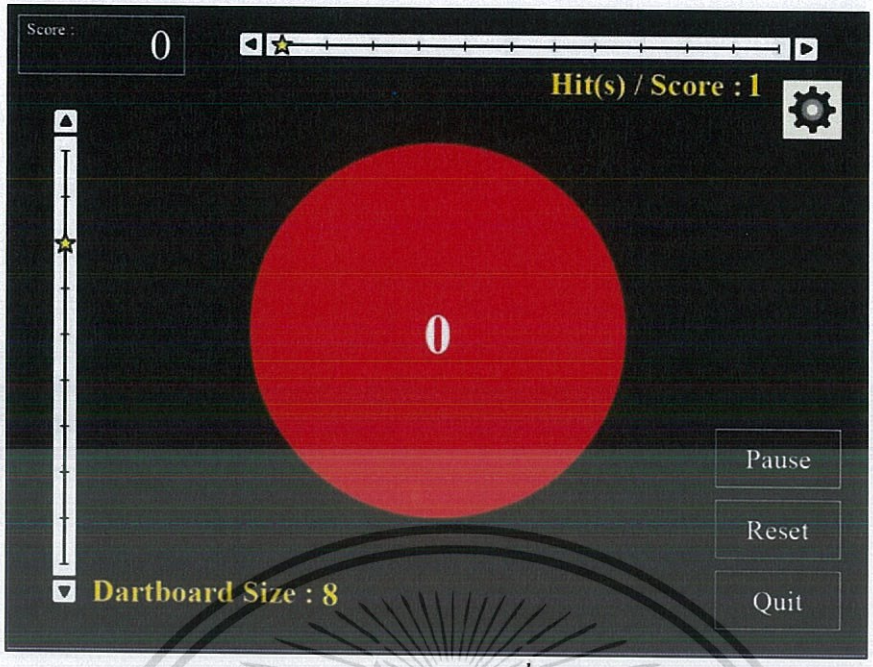
4.6.4.2 รายละเอียดของเกมที่ 2 “ เกมปาเป้า ”

หลังจากสร้างหน้าจอเกม และรายละเอียดเบื้องต้นของเกม ได้แก่ ช่องแสดงคะแนนด้านซ้ายบนของจอเกม แถบสำหรับปรับขนาดของเป้า หรือ Dartboard แถบสำหรับตั้งค่าว่าต้องปาลูกบอลให้โดนเป้ากี่ครั้งถึงจะได้ 1 คะแนน และปุ่มสำหรับตั้งค่า, Pause เกม, Reset เกม, ออกจากเกม โดยสามารถเลือกปรับค่าหรือกดปุ่มได้ด้วยการชี้ Laser Pointer รวมถึงเป้าสำหรับปาเป็นรูปทรงวงกลมสีแดง มีจำนวนครั้งที่บอลปาโดนเป้าบอกไว้ตรงกลางเป้า ดังภาพ 4.24

รูป 4.24 หน้าจอเกม “ เกมปาเป้า ” เบื้องต้น

เมื่อทดลองใช้ Laser Pointer ปรับขนาดของเป้าเป็นขนาดที่ 8 จะได้ดังภาพ 4.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



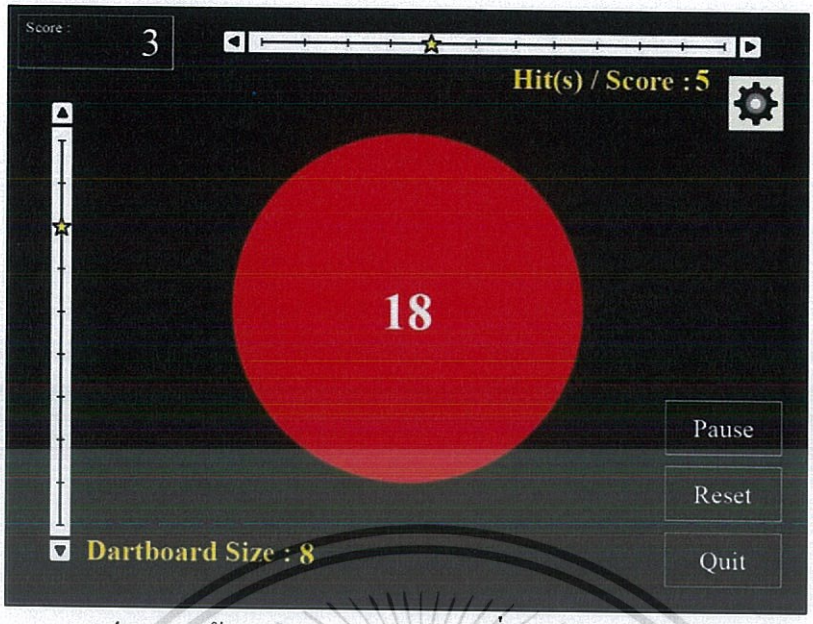
รูป 4.25 หน้าจอเกม " เกมปาเป้า " เมื่อปรับขนาดของเป้า

เมื่อใช้ Laser Pointer ปรับแถบสำหรับตั้งค่าว่าต้องปาลูกบอลให้โดนเป้ากี่ครั้งถึงจะได้ 1 คะแนน ตัวอย่างนี้ปรับให้ปาลูกบอลโดน 4 ครั้งได้ 1 คะแนน จะได้ดังภาพ 4.26

รูป 4.26 หน้าจอเกม " เกมปาเป้า " เมื่อปรับแถบ Hit(s) / Score

จากนั้นเมื่อปาลูกบอลโดนเป้า ตัวเกมจะนับคะแนนตามแถบที่ผู้ใช้ได้ปรับไว้ ตัวอย่าง ปาลูกบอลโดนเป้า 18 ครั้ง และคิดคะแนนเมื่อปาลูกบอลโดนเป้า 5 ครั้ง ดังนั้นปาลูกบอลโดนเป้า 18 ครั้ง จะคิดเป็น 3 คะแนน ดังรูป 4.27

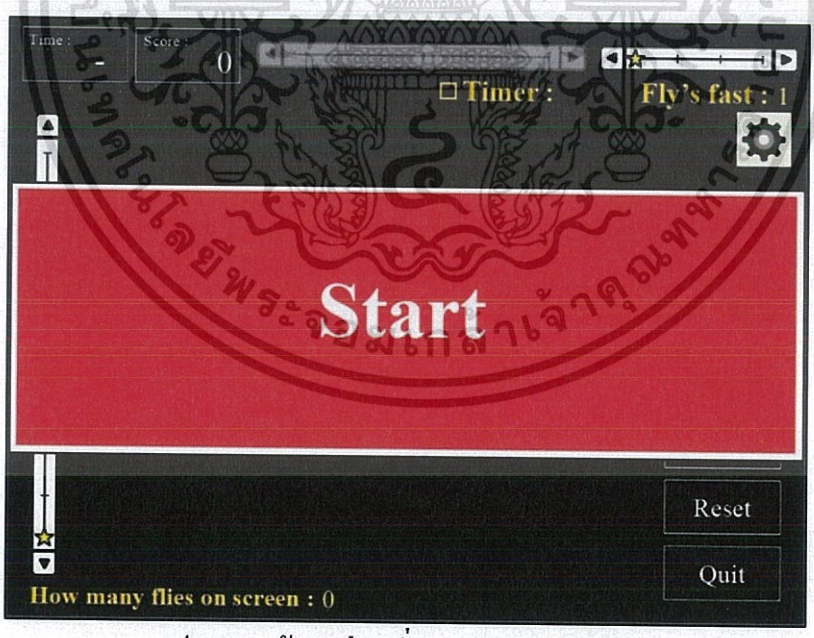
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.27 หน้าจอเกม " เกมปาเป้า " เมื่อปาลูกบอลโดนเป้า

4.6.4.3 รายละเอียดของเกมที่ 3 " เกมปาแมลงวัน "

เนื่องจากเกมปาแมลงวันมีการเคลื่อนไหวของแมลงวันอยู่ตลอดเวลา ดังนั้นเมื่อเข้าเกม จึงมีแถบ Start ให้ใช้ Laser Pointer ชี้เพื่อเริ่มเล่นเกม ดังรูป 4.28



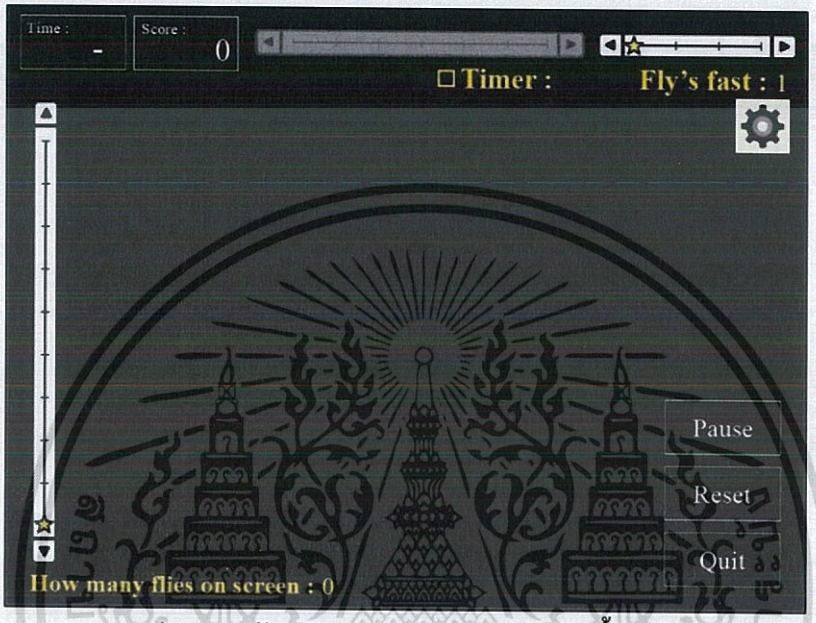
รูป 4.28 หน้าจอก่อนเริ่ม " เกมปาแมลงวัน "

ผู้จัดทำได้สร้างหน้าจอเกมและรายละเอียดเบื้องต้นต่างๆ ของเกมไว้ ได้แก่ ช่องแสดงเวลา ช่องแสดงคะแนนด้านซ้ายบนของจอเกม ช่องเปิดปิดการนับเวลา แถบตั้งค่าการนับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

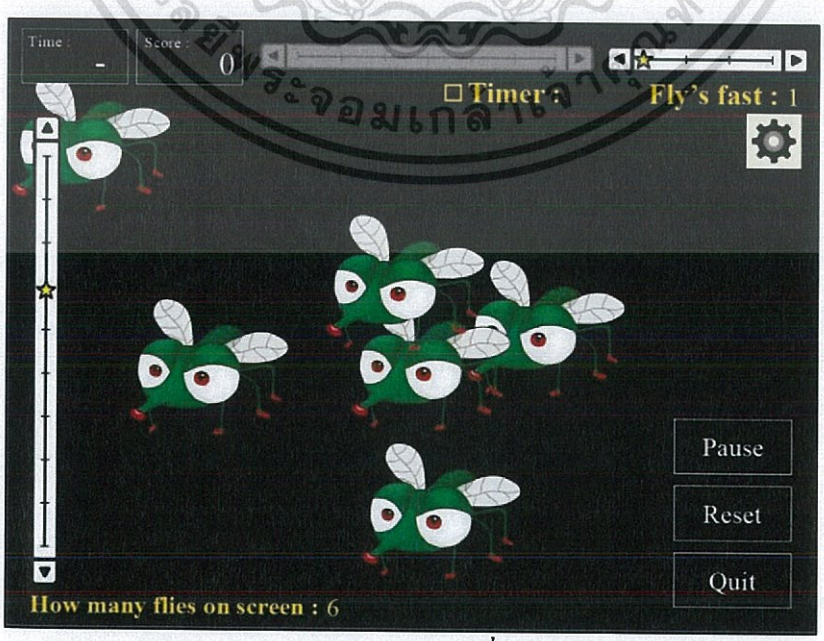
แถบตั้งค่าความเร็วมากที่สุดของแมลงวัน แถบปรับจำนวนแมลงวันบนหน้าจอ และปุ่มสำหรับตั้งค่า, Pause เกม, Reset เกม, ออกจากเกม

หลังจากใช้ Laser Pointer ชี้ที่แถบ Start เพื่อเริ่มเกม สามารถใช้ Laser Pointer ชี้ที่แถบด้านซ้ายเพื่อตั้งค่าจำนวนแมลงวัน จะมีรูปแมลงวันจำนวนตามที่ตั้งค่าไว้ บินด้วยความเร็วแบบสุ่มบนจอเกม ดังภาพ 4.29



รูป 4.29 หน้าจอเกม " เกมปาแมลงวัน " เบื้องต้น

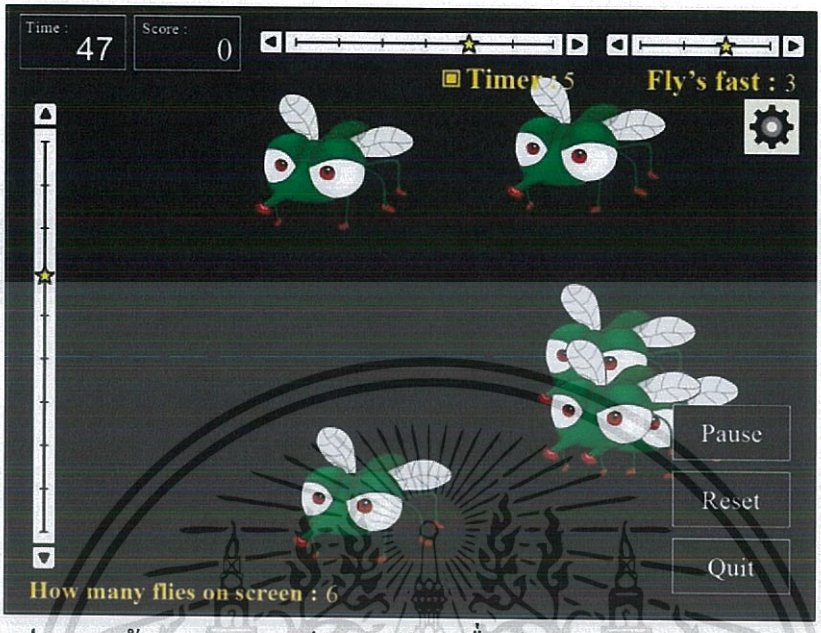
เมื่อปรับจำนวนแมลงวันบนจอเกมให้มี 6 ตัว จะได้ดังภาพ 4.30



รูป 4.30 หน้าจอเกม " เกมปาแมลงวัน " เมื่อปรับแถบจำนวนแมลงวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปิดการนับเวลา และปรับแถบการนับเวลาให้อยู่ในระดับ 5 คือนับเวลาถอยหลัง 50 วินาที จะได้ดังภาพ 4.31



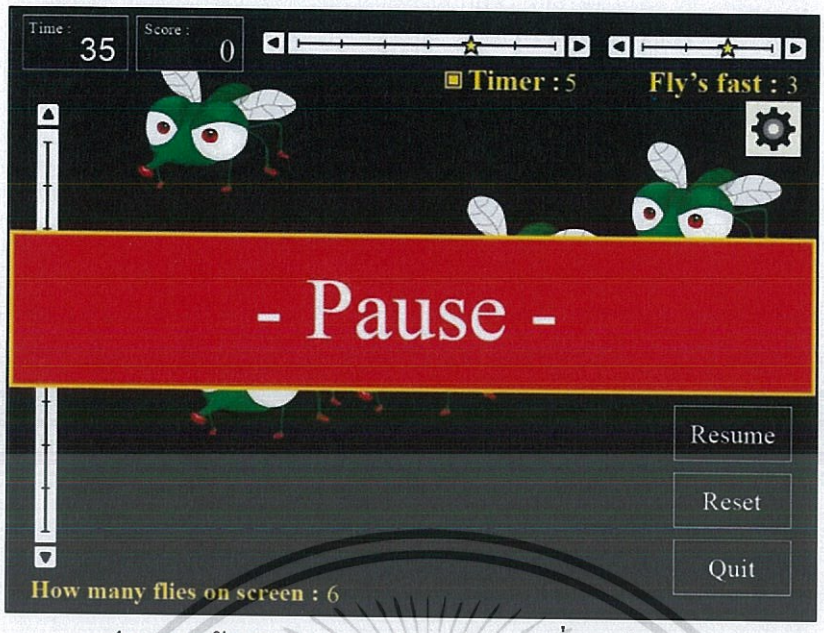
รูป 4.31 หน้าจอเกม " เกมปาแมลงวัน " เมื่อเปิดและปรับแถบตัวนับเวลา

เมื่อโยนลูกบอลโดนแมลงวัน รูปแมลงวันจะหายไปและจะนับคะแนน ดังภาพ 4.32

รูป 4.32 หน้าจอเกม " เกมปาแมลงวัน " เมื่อโยนลูกบอลโดนแมลงวัน

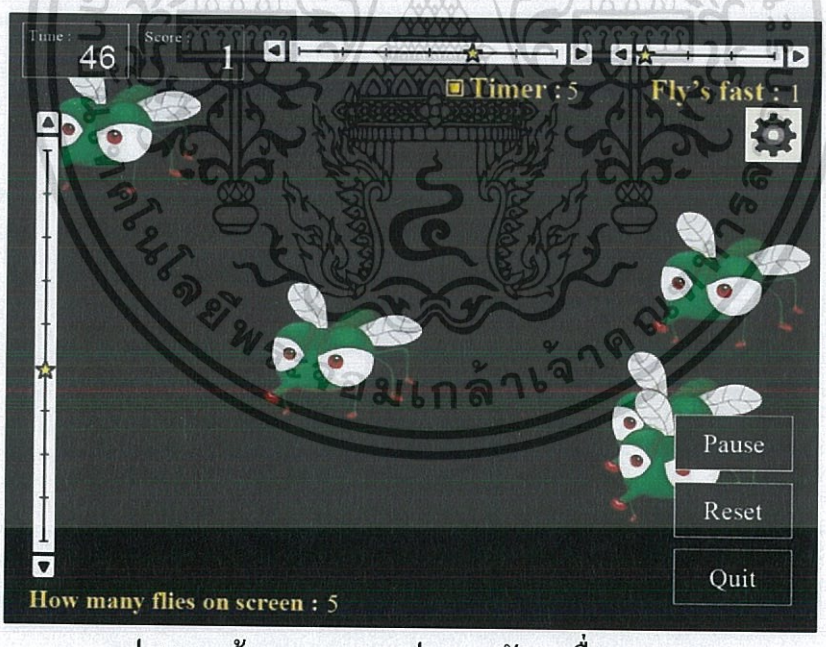
เมื่อใช้ Laser Pointer ชี้ที่ปุ่ม Pause แมลงวันจะหยุดการเคลื่อนไหว และมีแถบบอกสถานะ Pause ขึ้นมาที่กลางหน้าจอ ดังภาพ 4.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.33 หน้าจอเกม " เกมปาแมลงวัน " เมื่อกดปุ่ม Pause

เมื่อการตั้งค่าการนับเวลาเปิด และเวลาหมด แถบบอกสถานะ Time Up จะปรากฏขึ้นมาที่กลางหน้าจอ ดังภาพ 4.34



รูป 4.34 หน้าจอเกม " เกมปาแมลงวัน " เมื่อหมดเวลา

4.6.5 ผลการทดลอง

Pyglet Library สามารถนำมาใช้สร้างตัวเกมเพื่อใช้งานร่วมกับส่วนตรวจจับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปและบทวิจารณ์

เกมปาลูกบอลเป็น Platform ที่ใช้ในการเล่นเกมโดยการโยนลูกบอลใส่กำแพงที่ได้ฉายภาพเกมไว้อยู่ เป็นเกมที่เหมาะกับเด็กๆ เพื่อเสริมสร้างทักษะทั้งทางด้านร่างกายและความคิด สร้างโดยใช้หลักการของ Image Processing ในการวิเคราะห์ตำแหน่งของลูกบอลและการกระทบผนังของลูกบอล เพื่อนำข้อมูลที่ได้ไปใช้ในการทำงานของเกมต่างๆ

5.2 ปัญหาและอุปสรรค

1) สีของลูกบอลที่ใช้ได้มีจำกัด
2) ต้องใช้พื้นที่ในการทดลองค่อนข้างมาก
3) แสงไฟมีผลทำให้เกิดความคลาดเคลื่อนของการตรวจจับบอลภายในภาพ
4) ถ้ามีวัตถุในฉากที่เป็นทรงกลมและมีสีเดียวกับบอลที่กำหนดไว้ ระบบจะตรวจจับว่าเป็นบอล

5.3 แนวทางแก้ไขปัญหาและอุปสรรค

1) พัฒนาระบบให้ตรวจจับสีของลูกบอลได้ทุกสี
2) ทดลองหาขนาดของลูกบอลและพื้นที่ที่เหมาะสมในการเล่น
3) จัดการตำแหน่งที่วางกล้อง และสถานที่เล่นให้ดีเพื่อควบคุมแสงและตั้งค่าสีลูกบอลใหม่ทุกครั้งที่เปลี่ยนสถานที่เล่น

5.4 แนวทางการพัฒนาต่อ

1) เขียนเกมใหม่ๆมาลงเพิ่ม
2) เพิ่มประสิทธิภาพการตรวจจับด้วยการเพิ่มจำนวนกล้องเป็น 2 ตัว
3) ตรวจจับและกำหนดสีของลูกบอลที่นำมาใช้เล่นโดยอัตโนมัติ
4) ใช้หน่วยประมวลผลกราฟิกมาช่วยในการประมวลผล
5) ทำระบบ Multi-Player

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Jan Erik Solem. 2555. **Programming Computer Vision with Python.**
New York : Jan Erik Solem.

Ian Sommerville. 2554. **Software Engineering.** 9th ed. New York :
Addison-Wesley.

Bbee (alias). 2551. **รู้จักกับภาษา Python.** [Online].
Available : http://bbee.exteen.com/20080313/python.

Fraxel, Pctroll (alias). 2555. **Python คืออะไร.** [Online].
Available : http://www.mindphp.com/คู่มือ/73-คืออะไร/2417-python คืออะไร.html

Machinize (alias). 2558. **เริ่มต้น Computer Vision ง่ายๆ กับ OpenCV (ตอนที่1).** [Online].
Available : http://machinize.blogspot.com/2009/09/computer-vision-opencv.html.

Assoc. Prof. Dr. Montri Karnjanadecha. 2544. **บทที่ 1 แนะนำสู่การประมวลผลภาพดิจิตอล.** [Online].
Available : http://fivedots.coe.psu.ac.th/~montri/Teaching/image/chap1.htm.

Pichayanart Reerak. 2555. **ระบบสี (Color Model).** [Online].
Available : http://tc.mengrai.ac.th/krunoom/instruction/photoshop/unit1-2.php.

Wikipedia (alias). 2558. **การประมวลผลภาพ** [Online].
Available : https://th.wikipedia.org/wiki/การประมวลผลภาพ.

Silllovely (alias). 2556. **เทคโนโลยีการประมวลผลภาพ (Image processing).** [Online].
Available : https://silllovely.wordpress.com/2013/06/11/เทคโนโลยีการประมวลผลภา/

Opencv dev team. 2011-2014. **Feature Detection – OpenCV 2.4.12.0 documentation.** [Online].
Available : http://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?high
light=houghcircles.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Opencv dev team. 2011-2014. **Operations on Arrays – OpenCV 2.4.12.0 documentation.** [Online]. Available : http://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=houghcircles.

Big Vision LLC. 2015. **Learn OpenCV by Examples : Hough circle detection.** [Online]. Available : http://opencvexamples.blogspot.com/2013/10/hough-circle-detection.html.

Big Vision LLC. 2015. **Learn OpenCV by Examples.** [Online]. Available : http://opencvexamples.blogspot.com/.

Alexandre Poltorak, Andy Spencer. 2010. **OpenCV Tennis balls recognizing tutorial.** [Online]. Available : http://wiki.elphel.com/index.php?title=OpenCV_Tennis_balls_recognizing_tutorial.

Fraxel, Pctroll. 2012. **What are the correct usage/parameter values for HoughCircles in OpenCV for Iris detection?** [Online]. Available : http://stackoverflow.com/questions/10716464/.

Eugene Khvedchenya. 2014. **How to detect circles in noisy images.** [Online]. Available : http://computer-vision-talks.com/articles/how-to-detect-circles-in-noisy-image/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้