โต๊ะอัจฉริยะ
SMART TABLE

ชนน รังสิกวานิช
CHNON RUNGSIKAVANICH

ณิชา วงศ์สวัสดิวัฒนา
NICHA WONGSAWADDIWATTANA

ติมานนท์ หล่อสุพรรณพร
TIMANON LOSUPANPHORN

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

โต๊ะอัจฉริยะ
SMART TABLE

ชนน รังสิกวานิช
CHNON RUNGSIKAVANICH

ณิชา วงศ์สวัสดิวัฒนา
NICHA WONGSAWADDIWATTANA

ติมานนท์ หล่อสุพรรณพร
TIMANON LOSUPANPHORN

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

SMART TABLE

CHNON RUNGSIKAVANICH

NICHA WONGSAWADDIWATTANA

TIMANON LOSUPANPHORN

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

หัวข้อปริญญานิพนธ์	โต๊ะอัจฉริยะ			
รายชื่อนักศึกษา	นายชนน	รังสิกวานิช	รหัสนักศึกษา	53010289
	นางสาวณิชา	วงศ์สวัสดิวัฒนา	รหัสนักศึกษา	53010527
	นายติมานนท์	หล่อสุพรรณพร	รหัสนักศึกษา	53010553
ปริญญา	วิศวกรรมศาสตรบัณฑิต			
สาขาวิชา	วิศวกรรมสารสนเทศ			
พ.ศ.	2556			
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.พนารัตน์ เชิญถนอมวงศ์			

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

.....

(ผศ.ดร.พนารัตน์ เชิญถนอมวงศ์)
อาจารย์ผู้ควบคุมปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	โต๊ะอัจฉริยะ			
รายชื่อนักศึกษา	นายชนน	รังสิกวานิช	รหัสนักศึกษา	53010289
	นางสาวณิชา	วงศ์สวัสดิวัฒนา	รหัสนักศึกษา	53010527
	นายติมานนท์	หล่อสุพรรณพร	รหัสนักศึกษา	53010553
ปริญญา	วิศวกรรมศาสตรบัณฑิต			
สาขาวิชา	วิศวกรรมสารสนเทศ			
พ.ศ.	2556			
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.พนารัตน์ เชิญถนอมวงศ์			

บทคัดย่อ

ปัจจุบันเทคโนโลยีหน้าจอสัมผัสได้มีการพัฒนาไปอย่างมาก ซึ่งหน้าจอสัมผัสที่มีขนาดใหญ่จะใช้งานได้หลากหลายเป็นประโยชน์ในการนำเสนองาน ตลอดจนเป็นสื่อการเรียนการสอนที่น่าสนใจอีกรูปแบบหนึ่ง ซึ่งโครงงานนี้ได้เล็งเห็นถึงประโยชน์ของหน้าจอสัมผัสขนาดใหญ่นี้ แต่หน้าจอสัมผัสขนาดใหญ่ในปัจจุบันยังคงมีราคาที่ค่อนข้างสูง ทำให้โครงงานนี้ให้เลือกหาเทคโนโลยีหน้าจอสัมผัสด้วยต้นทุนที่ต่ำลง ซึ่งเทคโนโลยีที่ใช้คือเทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรด โดยใช้เทคนิค Rear Diffuse Illumination (Rear DI) ซึ่งเป็นการระบุตำแหน่งการสัมผัสโดยใช้กล้องเพื่อจับภาพความต่างของแสง ทำให้โต๊ะอัจฉริยะสามารถสัมผัสได้หลายตำแหน่งบนหน้าจอทำให้สามารถพัฒนาแอพพลิเคชั่นได้หลากหลาย โดยโครงงานนี้ได้นำเสนอแอพพลิเคชั่นวาดเขียนที่ผู้ใช้สามารถทำการวาดเขียนลงบนหน้าจอได้ สามารถนำไปประยุกต์ใช้ในการนำเสนอผลงาน หรือเป็นสื่อการเรียนการสอนได้ และแอพพลิเคชั่นที่นำเสนองานเฉพาะทางในด้านการตกแต่งภายใน ที่ช่วยอำนวยความสะดวกในการนำเสนอการจัดวางฟอร์นิเจอร์ต่างๆได้ นอกจากนี้แล้วโครงงานนี้ยังสามารถต่อยอดแอพพลิเคชั่นต่างๆ เพื่อนำมาประยุกต์ใช้ในชีวิตประจำวันเด้อีกมากมาย

Thesis Title	Smart Table		
Student	Mr. Chnon	Rungsikavanich	Student ID. 53010289
	Miss Nicha	Wongsawaddiwattana	Student ID. 53010527
	Mr. Timanon	Losuoanporn	Student ID. 53010553
Degree	Bachelor of Engineering		
Program	Information Engineering		
Year	2013		
Thesis Advisor	Asst.Prof.Dr. Panarat	Cherntanomwong	

ABSTRACT

Nowadays, touch screen technology has been developed rapidly. The big touch screen has been realized that it is useful in presenting and learning. However, it is still expensive. Therefore, this thesis proposes the development of the big and smart touch screen using the infrared technology with the Rear Diffused Illumination (Rear DI) method. With Rear DI method, infrared light is shined at the screen from below the touch surface. When the object touches the screen, the more light is reflected and detected by camera. This touch screen is applicable for multi-user with multi-touch screen. Two applications are developed; (i) Painting application in which the user can draw, edit, and erase on the screen. This application makes the presentation more interesting, (ii) Interior Design application which is the specific application for the interior designer. It can help user facilitate placement of furniture. Although, we developed only two applications, there are many applications that can be developed for daily life applications.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เป็นผลงานที่สำคัญอย่างยิ่งสำหรับผู้จัดทำ เนื่องจากผู้จัดทำมีความมุ่งมั่นในการสร้างสรรค์ผลงานให้ออกมาเป็นรูปธรรมซึ่งการทำส่วนต่างๆนั้นไม่สามารถทำเพียงผู้เดียวได้แต่ต้องอาศัยการทำงานเป็นทีม การความช่วยเหลือจากผู้รู้

คณะผู้จัดทำขอขอบพระคุณท่านอาจารย์ที่ปรึกษาปริญญานิพนธ์ ผศ.ดร.พนารัตน์ เชิญถนอมวงศ์ ที่มีความเมตตาเสียสละเวลาในการช่วยเหลือดูแลคอยให้คำปรึกษาด้านต่างๆอย่างเป็นกันเอง ทั้งการแนะนำในตัวชิ้นงานและในการจัดทำปริญญานิพนธ์ฉบับนี้ จนโครงงานชิ้นนี้สามารถสำเร็จได้

ขอขอบพระคุณเพื่อนๆทุกๆคนทั้งในคณะสาขาวิชาเดียวกันและต่างสาขา และเพื่อนคณะสถาปัตยกรรมที่ให้คำแนะนำช่วยเหลือในการจัดทำชิ้นงาน

ขอขอบพระคุณบิดา มารดา ที่ช่วยเหลือสนับสนุนในการหาอุปกรณ์ต่างๆเพื่อจัดทำชิ้นงาน ให้คำแนะนำติชม และเป็นกำลังใจสำคัญในการทำโครงงานชิ้นนี้จนสำเร็จ

สุดท้ายนี้ต้องขอขอบพระคุณท่านอาจารย์ทุกๆท่านที่ให้วิชาความรู้ที่สามารถนำมาประยุกต์ใช้กับโครงงานชิ้นนี้และยังสามารถนำความรู้ที่ได้รับไปประกอบอาชีพต่อไปในอนาคต

ชนน รังสิกวานิช
ณิชา วงศ์สวัสดิวัฒนา
ติมานนท์ หล่อสุพรรณพร

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VI
สารบัญรูป VII
บทที่ 1 บทนำ 1
1.1 ความสำคัญและที่มาของปัญหา 1
1.2 จุดประสงค์ 2
1.3 การทำงานโดยรวมของระบบ 2
1.4 ขอบเขตของโครงงาน 5
1.5 ผลที่คาดว่าจะได้รับ 5
1.6 อุปกรณ์ที่ต้องใช้ 5
1.7 ขั้นตอนการดำเนินงาน 6
บทที่ 2 ทฤษฎีพื้นฐาน 7
2.1 ส่วนฮาร์ดแวร์ 7
2.1.1 หน้าจอสัมผัสโดยใช้เทคโนโลยีอินฟราเรด 7
2.1.2 กล้องเว็บแคมอินฟราเรด 12
2.1.3 การแสดงผลของหน้าจอสัมผัสแบบอินฟราเรดโดยใช้เครื่องฉายภาพ 13
2.1.4 กฎการสะท้อน 14
2.2 ส่วนซอฟต์แวร์ 16
2.2.1 โปรแกรม Community Core vision (CCV) 16
2.2.2 โปรโตคอล TUIO 18
2.2.3 โปรแกรม Touch Injector 19
2.2.4 ภาษาจาวา 20
บทที่ 3 การออกแบบ 24
3.1 อุปกรณ์ฮาร์ดแวร์ 24
3.1.1 กล้องเว็บแคมอินฟราเรด 24
3.1.2 การแสดงผลของหน้าจอสัมผัสแบบอินฟราเรดโดยใช้เครื่องฉายภาพ 25
3.1.3 หลอดแอลอีดีอินฟราเรด 26

สารบัญ (ต่อ)

หน้า

3.2 การออกแบบฮาร์ดแวร์ 28
3.2.1 โครงสร้างของโต๊ะอัจฉริยะ 28
3.2.2 หน้าจอสัมผัสเทคโนโลยีอินฟราเรด เทคนิค Rear Diffused Illumination 29
3.2.3 การสะท้อนเครื่องฉายภาพ 30
3.2.4 การสะท้อนกล้องเว็บแคมอินฟราเรด 32
3.2.5 โครงสร้างโดยรวมของฮาร์ดแวร์ 33
3.3 การออกแบบซอฟต์แวร์ 34
3.3.1 การตั้งค่าเพื่อการระบุตำแหน่งการสัมผัสหน้าจอ 34
3.3.2 ซอฟต์แวร์แอพพลิเคชั่นที่สามารถใช้งานร่วมกับ Microsoft Windows 8 36
3.3.3 ซอฟต์แวร์แอพพลิเคชั่นที่ทำงานร่วมกับภาษาจาวาเบื้องต้น 36
บทที่ 4 ผลการทดลอง 56
4.1 ผลการทำการคาลิเบรชั่น 56
4.2 ผลการเชื่อมต่อโปรแกรม CCV กับ Touch Injector 57
4.3 ผลการทดลองผลการเชื่อมต่อโปรแกรม CCV กับ แอพพลิเคชั่นที่เขียนด้วยภาษาจาวา 60
4.3.1 ผลการทดลองหน้าเมนูหลัก 61
4.3.2 ผลการทดลองแอพพลิเคชั่นตกแต่งภายในบ้าน 65
4.3.3 ผลการทดลองแอพพลิเคชั่นวาดเขียน 67
4.4 ผลการทดลองตามขอบเขตที่กำหนด 70
บทที่ 5 สรุปและแนวทางการแก้ไข 75
5.1 สรุปผลการทำงาน 75
5.2 ปัญหาระหว่างการดำเนินงาน 75
5.3 แนวทางการแก้ไข 75
5.4 แนวทางการพัฒนาต่อ 76
บรรณานุกรม 77
ภาคผนวก 78

สารบัญตาราง

ตารางที่ หน้า
1.1 ระยะเวลาการดำเนินงาน 6
2.1 ข้อดีข้อเสียของเทคนิค FTIR 11
2.2 ข้อดีข้อเสียของเทคนิค Rear DI 11
2.3 ข้อดีข้อเสียของเทคนิค Front DI 11
2.4 ข้อดีข้อเสียของเทคนิค LLP 11
2.5 ข้อดีข้อเสียของเทคนิค DSI 12
2.6 ข้อดีข้อเสียของเทคนิค LED-LP 12
2.7 เปรียบเทียบข้อดีข้อเสียของเครื่องฉายภาพแบบ LCD และ DLP 14

สารบัญรูป

รูปที่ หน้า

1.1 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค Rear DI.....3
1.2 เทคโนโลนยีหน้าจอสัมผัสแบบอินฟราเรดโดยใช้เทคนิคแบบ Rear DI.....3
1.3 กล้องอินฟราเรดจับภาพจุดที่มีการสัมผัสเป็นจุดที่มีความเข้มของแสงอินฟราเรดอยู่มากกว่าจุดที่ไม่ได้สัมผัส ...4
1.4 แอพพลิเคชั่นที่สามารถวาดรูปได้.....4
2.1 รูปที่ได้จากกล้องอินฟราเรดเมื่อมีการสัมผัส.....7
2.2 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค FTIR.....8
2.3 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค LLP.....8
2.4 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค DSI.....9
2.5 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค LED – LP 9
2.6 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค Front DI.....10
2.7 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค Rear DI.....10
2.8 หลักการทำงานของเครื่องฉายภาพแบบ LCD.....13
2.9 หลักการทำงานของเครื่องฉายภาพแบบ DLP.....14
2.10 การสะท้อนแสงของพื้นที่ผิวเรียบและขรุขระ.....15
2.11 การสะท้อนของกระจกเงา.....15
2.12 โปรแกรม Community Core Vision (CCV).....16
2.13 หลักการเบื้องต้นของโปรแกรม CCV.....16
2.14 การใช้งานโปรแกรม Community Core Vision (CCV).....17
2.15 หลักการทำงานของโปรโตคอล TUIO.....18
2.16 โปรแกรม Touch Injector.....19
2.17 หลักการทำงานของ Touch Injector.....20
2.18 จาวา.....20
2.19 ภาษาจาวาเบื้องต้น (1).....22
2.20 ภาษาจาวาเบื้องต้น (2).....23
3.1 กล้อง Logitech C170.....24
3.2 แผ่น Floppy Disk.....24
3.3 การนำส่วนที่เป็นฟิลเตอร์กรองแสงอินฟราเรดออกและใส่ฟิลเตอร์ที่กรองเฉพาะแสงอินฟราเรด (IR-pass filter) เข้าไปแทนที่.....25
3.4 ภาพที่เกิดจากการรับภาพจากเทคโนโลยีหน้าจอสัมผัสอินฟราเรด เทคนิค Rear DI.....25
3.5 เครื่องฉายภาพ EPSON BusinessEB - S02.....26

3.6 คุณสมบัติของหลอดแอลอีดีอินฟราเรดรุ่น TOIR -50b94 bdEa 27
3.7 โครงสร้างของโต๊ะอัจฉริยะ 28
3.8 การวางแอลอีดีอินฟราเรด 29
3.9 การวางแอลอีดีอินฟราเรด 30
3.10 ระยะฉายภาพ.....31
3.11 การสะท้อนภาพจากเครื่องฉายภาพโดยใช้กระจกเงาเพื่อเป็นการเพิ่มระยะให้สามารถฉายภาพได้.....31
3.12 มุมการสะท้อนเครื่องฉายภาพโดยใช้กระจกเงาเมื่อมองภาพตัดขวาง 32
3.13 การสะท้อนการรับภาพของกล้องเว็บแคมอินฟราเรดเมื่อมองภาพตัดขวาง 32
3.14 แบบโครงสร้างภายในของโต๊ะอัจฉริยะ 33
3.15 โครงสร้างภายในของโต๊ะอัจฉริยะ.....33
3.16 แบบโดยรวมของโต๊ะอัจฉริยะ 34
3.17 การปรับค่าในโปรแกรมccv เพื่อให้ได้ความแม่นยำในการสัมผัส.....34
3.18 การแสดงการปรับค่าที่ได้ความแม่นยำเมื่อมีการสัมผัส.....35
3.19 การทำการคาลิเบรชั่นเพื่อให้การสัมผัสตรงกับหน้าจอที่ใช้งานจริง 35
3.20 บล็อคไดอะแกรมการทำงานของซอฟต์แวร์ ที่สามารถใช้งานร่วมกับ Microsoft Windows8 36
3.21 บล็อคไดอะแกรมการทำงานของซอฟต์แวร์ ที่สามารถใช้งานร่วมกับแอพพลิเคชั่นภาษาจาวา 37
3.22 แบบหน้าจออินเตอร์เฟสของแอพพลิเคชั่น 37
3.23 คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น 38
3.24 คลาส Scene.....38
3.25 ไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่น.....39
3.26 การใช้คำสั่ง extends.....39
3.27 ส่วนของการสร้างหน้าต่าง (frame) และการใส่ภาพพื้นหลัง.....40
3.28 ส่วนของการกดปุ่มจะแสดงลิสให้เห็น 41
3.29 ส่วนของการสร้างเซลล์ให้ลิส เมื่อมีการกดที่ลิสจะทำการเปลี่ยนภาพพื้นหลัง 42
3.30 การสร้างเซลล์ให้ลิส เมื่อมีการกดที่ลิสจะทำการเปลี่ยนภาพพื้นหลัง(ต่อ).....42
3.31 การสร้างปุ่มกดเมื่อกดแล้วรูปไฟล์ .svg จะแสดง 43
3.32 ส่วน MTHandler.....43
3.33 แอพพลิเคชั่นวาดเขียน 44
3.34 คลาสเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น.....44
3.35 ส่วนของคลาส Scene 45
3.36 ส่วนของคลาส Scene (ต่อ).....46
3.37 ส่วนของคลาส Scene (ต่อ).....47
3.38 เมธอดของคลาส Scene ที่สร้างรูปเมื่อสัมผัส.....47
3.39 ไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่น.....48

3.40 การสร้างหน้าต่างสำหรับการวาดเขียน 48
3.41 การล้างหน้าจอทั้งหมด 49
3.42 ส่วนของการลบและการเขียน 49
3.43 เมธอด Texture 50
3.44 แบบหน้าจอเมนูหลัก 50
3.45 คลาสเริ่มต้นในการเรียกใช้แอพพลิเคชั่น 51
3.46 คลาส Scene 51
3.47 ไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่น 52
3.48 การสร้างหน้าต่างของแอพพลิเคชั่น 52
3.49 การสร้างลิส 53
3.50 การนำไอคอนแอพพลิเคชั่นใส่ในลิส 53
3.51 การคลิกจะสร้าง Scene ขึ้นมา 54
3.52 รูปแบบหน้าต่าง Scene ที่แสดงขึ้นมา 55
4.1 การสัมผัสหน้าจอที่ยังไม่ได้ทำการคาลิเบรชั่น 56
4.2 การสัมผัสหน้าจอที่ทำการคาลิเบรชั่นเรียบร้อย 56
4.3 การใช้โต๊ะอัจฉริยะร่วมกับ Microsoft Windows8 57
4.4 การแสดงการพิมพ์บนคีย์บอร์ดหน้าจอสัมผัสได้ 57
4.5 การแสดงการใช้งานแทนเม้าส์ โดยใช้งานได้กับทุกโปรแกรมบน Windows 8 58
4.6 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การเล่นเกมส์ออนไลน์ 58
4.7 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การเล่น Facebook 59
4.8 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การค้นหาบน Google map 59
4.9 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การใช้โปรแกรมไลน์ (Line) 60
4.10 การสัมผัสขยายตัวอักษรที่เขียนด้วยภาษาจาวา 60
4.11 การสัมผัสย่อตัวอักษรที่เขียนด้วยภาษาจาวา 61
4.12 การแสดงการเลื่อนซ้ายขวาเพื่อเลือกแอพพลิเคชั่น 61
4.13 การแสดงการเลือกที่ไอคอนแอพพลิเคชั่นั้นจะแสดงเป็นหน้าต่าง 62
4.14 หน้าต่างแอพพลิเคชั่นสามารถทำการเคลื่อนย้ายได้ 62
4.15 การแสดงความสามารถย่อขยายหน้าต่างได้ 63
4.16 การแสดงความสามารถเลือกแอพพลิเคชั่นขึ้นมาอีกได้ 63
4.17 การแสดงความสามารถทำงานได้ทุกแอพพลิเคชั่นที่เลือกขึ้นมา 63
4.18 การปิดแอพพลิเคชั่นที่เลือกขึ้นมากดปุ่ม X 64
4.19 การขยายแอพพลิเคชั่นที่เลือกให้เต็มหน้าจอกดปุ่มสี่เหลี่ยม 64
4.20 การปิดโปรแกรมเต็มหน้าจอให้กดตรงมุมขวาแล้วลากมากดปุ่ม X จะกลับสู่หน้าเมนูหลัก 64
4.21 การเลือกรูปแบบบ้าน 65

4.22 การแสดงความสามารถเลือกฟอนิเจอร์ โต๊ะทำงาน โซฟา เตียง ตู้ เตาแก๊ส โต๊ะทานข้าว และรถยนต์.....65
4.23 ฟอร์นิเจอร์ต่างๆสามารถลากย้ายที่ได้.....66
4.24 ความสามารถในการย่อขยายฟอร์นิเจอร์ได้โดยใช้สองนิ้วกดแล้วลากได้.....66
4.25 ต้องการลบฟอร์นิเจอร์ให้กดที่รูปฟอร์นิเจอร์ที่ต้องการลบค้างไว้จนวงกลมสีแดงวิ่งเต็มวง.....67
4.26 เมื่อกด + จะเปลี่ยนแบบบ้านที่แสดง.....67
4.27 การเขียนสามารถเลือกสีของลายเส้นได้ 3 สีด้วยกันคือสีน้ำเงิน สีเขียว และสีแดง.....68
4.28 ความสามารถในการเขียนได้หลายคน.....68
4.29 การลบทีละจุดเลือกที่ยางลบ แล้วลากไปที่บริเวณที่ต้องการจะลบ.....69
4.30 การล้างหน้าจอทั้งหมดให้เลือกแปรง.....69
4.31 การใช้ทัชคีย์บอร์ดกับโต๊ะอัจฉริยะ.....70
4.32 การใช้โต๊ะอัจฉริยะในการนำเสนอผลงาน.....71
4.33 แอพพลิเคชั่นที่สามารถรองรับการสัมผัสได้มากกว่าหนึ่งจุด.....71
4.34 แอพพลิเคชั่นที่สามารถรองรับการสัมผัสได้มากกว่าหนึ่งจุด.....72
4.35 แอพพลิเคชั่นที่สามารถรองรับการสัมผัสได้มากกว่าหนึ่งจุด.....72
4.36 หน้าจอเมนูหลัก.....73
4.37 แอพพลิเคชั่นออกแบบกตกแต่งภายในบ้าน.....73
4.38 แอพพลิเคชั่นวาดเขียน.....73

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทำงานโดยทั่วไปในปัจจุบันเน้นการใช้เอกสารที่เป็นกระดาษ ทำให้สิ้นเปลืองทรัพยากรเป็นจำนวนมาก และในปัจจุบันเทคโนโลยีคอมพิวเตอร์ รวมถึงหน้าจอต่างๆได้มีการพัฒนามากขึ้น ทำให้ทางผู้จัดทำสนใจนำเทคโนโลยีมาประยุกต์ใช้เพื่อให้การทำงานมีความสะดวกสบายมากยิ่งขึ้นและช่วยลดการใช้ทรัพยากรได้

ผู้จัดทำจึงได้จัดทำปริญญานิพนธ์เรื่อง โต๊ะอัจฉริยะ (Smart Table) โดยออกแบบให้สามารถเป็นหน้าจอสัมผัส (Touch Screen) ของคอมพิวเตอร์ได้ ซึ่งหน้าจอสัมผัสที่ใช้งานในปัจจุบันนั้นมีอยู่ 4 ประเภท คือ เทคโนโลยี Resistive, Capacitive, Surface-acoustic-wave (SAW) และอินฟราเรด (Infrared) ซึ่งเทคโนโลยีแบบ Resistive จะประกอบด้วย เลเยอร์ (Layer) ด้านบนที่ยืดหยุ่นและเลเยอร์ด้านล่างอยู่บนพื้นแข็ง และคั่นระหว่าง 2 เลเยอร์ด้วยเม็ดฉนวนซึ่งทำให้ 2 เลเยอร์นั้นไม่สัมผัสกัน เมื่อสัมผัสจะทำให้เลเยอร์ 2 เลเยอร์นั้นต่อถึงกัน ทำให้สามารถระบุตำแหน่งที่สัมผัสได้ ซึ่งข้อเสียของหน้าจอสัมผัสที่ใช้เทคนิค Resistive คือจะเกิดการเลื่อนของจุดได้ง่าย และ เกิดความเสียหายจากของมีคมได้ง่าย ประเภทที่สองเทคโนโลยี Capacitive นั้นมีคุณสมบัติโดดเด่นทั้งความทนทานและความโปร่งแสงมักเป็นที่นิยมในแอพพลิเคชั่น (Application) ประเภทเกมส์ เอทีเอ็ม โดยโครงสร้างของเทคโนโลยีนี้จะประกอบด้วยแผ่นแก้วเคลือบผิวออกไซต์ของโลหะแบบโปร่งแสง เมื่อใช้งานก็จะมีการป้อนแรงดันไฟฟ้าที่มุมทั้งสี่ของหน้าจอสัมผัสเพื่อสร้างสนามแม่เหล็กไฟฟ้า ผู้ใช้ต้องใช้มือเปล่าๆ สัมผัสที่หน้าจอเพื่อดึงกระแส ทำให้สามารถระบุตำแหน่งสัมผัสได้ ซึ่งข้อเสียของเทคโนโลยีนี้คือ ความไวของอุปกรณ์จะด้อยลงได้เมื่อถูกรบกวนจากสัญญาณอิเล็คทรอนิคส์ ใช้พลังงานมาก และราคาค่อนข้างสูง ประเภทที่สามคือเทคโนโลยี Surface-acoustic-wave (SAW) เป็นเทคโนโลยีที่มีความคมชัดสูง โครงสร้างของเทคโนโลยีนี้จะมีตัวส่งสัญญาณยึดติดกับขอบกระจกเพื่อส่งสัญญาณอัลตราโซนิกส์ (Ultrasonic) ไปทั้งสองระนาบ คลื่นนี้จะสะท้อนผ่านไปทั้งพื้นผิวของกระจกมายังเซ็นเซอร์ (Sensor) อีกด้านหนึ่งเมื่อมีการสัมผัสด้วยปลายนิ้วหรือ Stylus จะมีการดูดซับพลังงานจากคลื่นเสียงทำให้แผงควบคุมสามารถวัดตำแหน่งการสัมผัสได้ ซึ่งทั้งสามประเภทที่กล่าวมานั้นหากนำมาทำเป็นหน้าจอขนาดใหญ่ เช่น หน้าจอสำหรับโต๊ะ จะมีความยุ่งยากในการผลิต และมีราคาที่สูงมาก จากปัญหาดังกล่าวจึงมีแนวคิดที่ใช้เทคโนโลยีอินฟราเรด ซึ่งเป็นเทคโนโลยีหน้าจอสัมผัสที่มีความแม่นยำสูง สามารถใช้กับหน้าจอแสดงผลขนาดใหญ่ และมีราคาที่ต่ำกว่าเทคโนโลยีอื่น จึงนำเทคโนโลยีนี้มาใช้กับโต๊ะอัจฉริยะ

หน้าจอสัมผัสโดยใช้เทคโนโลยีอินฟราเรด ทำงานโดยการตรวจจับแสงอินฟราเรด (Infrared) และใช้หลักการกระจายของแสงเพื่อให้สามารถตรวจหาตำแหน่งที่สัมผัสได้ โดยเทคโนโลยีหน้าจอสัมผัสอินฟราเรดนี้มีเทคนิค 6 แบบด้วยกันคือ Frustrated Total Internal Reflection (FTIR), Rear Diffused Illumination (Rear DI), Laser Light Plane (LLP), Diffused Surface Illumination (DSI), LED Light Plane (LED-LP) และ Front Diffused Illumination (Front DI) โดยทางผู้จัดทำได้เลือกใช้เทคโนโลยีอินฟราเรดเทคนิคแบบ

Rear Diffused Illumination (Rear DI) เนื่องจากเป็นเทคนิคที่สามารถทำเป็นหน้าจอสำหรับโต๊ะขนาดใหญ่ได้และประสิทธิภาพสามารถยอมรับได้

ซอฟต์แวร์ (Software) ของโต๊ะอัจฉริยะสามารถตรวจจับการสัมผัสที่หน้าจอโดยใช้หลักการของการประมวลผลภาพ (Image processing) ซึ่งการตรวจจับการเคลื่อนไหวโดยวิธีนี้จะทำให้สามารถสัมผัสได้หลายๆรูปแบบในเวลาเดียวกัน (Multi-Touch Screen) ทำให้ใช้งานได้หลายรูปแบบ และเพิ่มความสะดวกสบายในการทำงานได้อีกด้วย

1.2 จุดประสงค์

- เพื่อออกแบบและสร้างระบบเทคโนโลยีสารสนเทศในการสนับสนุนการทำงานในปัจจุบันที่เน้นการใช้เทคโนโลยีคอมพิวเตอร์
- เพื่อศึกษาหลักการทำงานของหน้าจอสัมผัสเทคโนโลยีอินฟราเรดโครงสร้างแบบ Rear Diffused Illumination (Rear DI)
- เพื่อศึกษาการเขียนแอพพลิเคชั่นด้วยภาษา Java
- เพื่อศึกษาการประยุกต์ใช้หลักการ การประมวลผลภาพ (Image Processing)

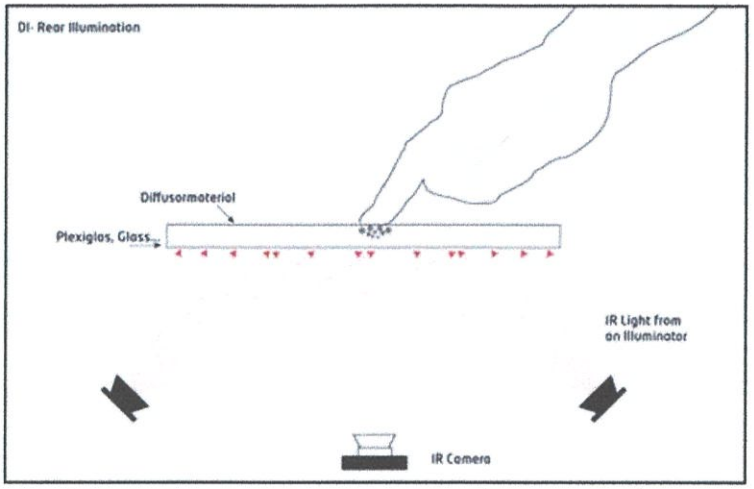
1.3 การทำงานโดยรวมของระบบ

ระบบนี้ประกอบด้วย 3 ส่วนหลักคือ

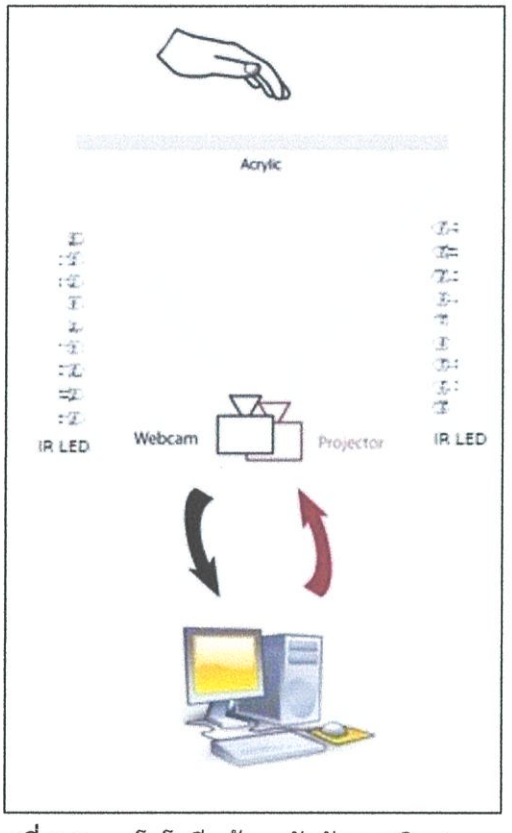
1) ส่วนของฮาร์ดแวร์ซึ่งคือส่วนที่เป็นโต๊ะ ที่ใช้เทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรด โดยใช้เทคนิคแบบ Rear Diffused Illumination (Rear DI)
2) ส่วนของซอฟต์แวร์ที่ทำหน้าที่ประมวลผลจากกล้อง เพื่อระบุตำแหน่งของนิ้วมือที่สัมผัสหน้าจอ
3) ส่วนของซอฟต์แวร์ที่เป็นแอพพลิเคชั่น

โดยสามารถอธิบายแต่ละส่วนได้ดังนี้

1) ส่วนของฮาร์ดแวร์ซึ่งคือส่วนที่เป็นโต๊ะ จะใช้เทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรด โดยใช้เทคนิคแบบ Rear Diffused Illumination (Rear DI) โดยหลักการทำงานคือการฉายแสงอินฟราเรดให้กระจายอยู่ทั่วทำให้เมื่อผู้ใช้สัมผัสหน้าจอจะเกิดการกระจายของแสงอินฟราเรด ดังรูปที่ 1.1 เมื่อนำกล้องอินฟราเรด (Infrared camera) มาจับภาพจะสามารถระบุตำแหน่งที่สัมผัสได้ และคอมพิวเตอร์จะประมวลผลและส่งออกไปยังเครื่องฉายภาพ ดังรูปที่ 1.2



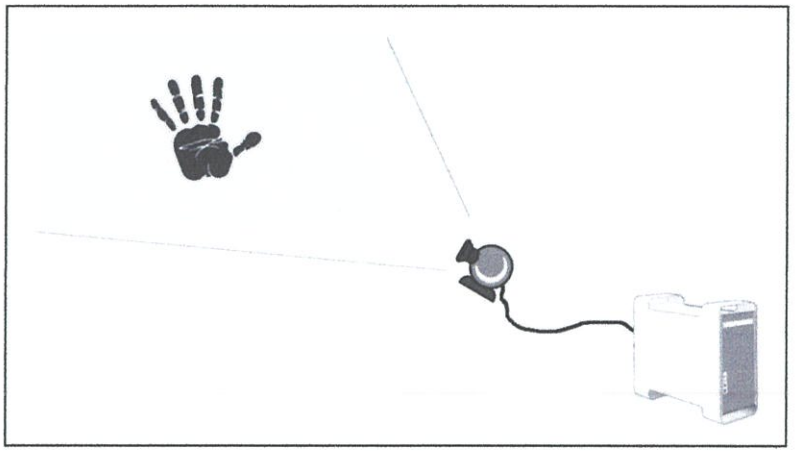
รูปที่ 1.1 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอ และชิ้นส่วนที่จำเป็นสำหรับเทคนิค Rear DI [1]



รูปที่ 1.2 เทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรด โดยใช้เทคนิคแบบ Rear Diffused Illumination (Rear DI)

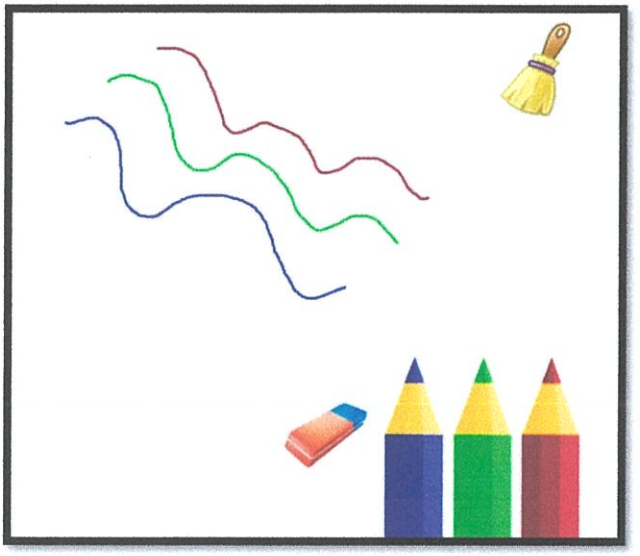
2) ส่วนของซอฟต์แวร์ที่ทำหน้าที่ประมวลผลจากกล้อง เพื่อระบุตำแหน่งของนิ้วมือที่สัมผัสหน้าจอ เมื่อมีการสัมผัสหน้าจอจะเกิดการกระจายของแสงอินฟราเรด เมื่อนำกล้องอินฟราเรดมาจับภาพ จะปรากฏจุดที่มีการสัมผัสเป็นจุดที่มีความเข้มของแสงอินฟราเรดอยู่มากกว่าจุดที่ไม่ได้สัมผัส ดัง

รูปที่ 1.3 เมื่อนำกล้องอินฟราเรดนั้นเชื่อมต่อกับความพิวเตอร์เพื่อประมวลผล โดยซอฟต์แวร์ที่เขียนขึ้นจะใช้หลักการของการประมวลผลภาพซึ่งเพื่อหาตำแหน่งที่สัมผัส และสามารถตอบสนองผู้ใช้งานได้



รูปที่ 1.3 กล้องอินฟราเรดจับภาพจุดที่มีการสัมผัสเป็นจุดที่มีความเข้มของแสงอินฟราเรดอยู่มากกว่าจุดที่ไม่ได้สัมผัส

3) ส่วนของซอฟต์แวร์ที่เป็นแอพพลิเคชั่น ที่สามารถใช้งานกับ Microsoft Windows 8 โดยใช้การสัมผัสเพื่อป้อนอินพุตต่างๆเช่น เม้าส์ (Mouse)และ คีย์บอร์ด (Keyboard) ทำให้มีความสามารถเป็นคอมพิวเตอร์หน้าจอสัมผัสได้ และมีการเขียนแอพพลิเคชั่นอื่นๆที่สามารถทำงานให้สามารถสัมผัสได้หลายๆรูปแบบในเวลาเดียวกัน (Multi-Touch Screen)ได้ เช่น แอพพลิเคชั่นที่สามารถวาดเขียนได้ ดังรูปที่ 1.4



รูปที่ 1.4 แอพพลิเคชั่นที่สามารถวาดรูปได้

1.4 ขอบเขตของโครงงาน

- สามารถเป็นหน้าจอสัมผัสของคอมพิวเตอร์
- สามารถใช้งานกับ Microsoft Windows 8 โดยใช้การสัมผัสเพื่อป้อนอินพุตต่างๆ ได้
- สามารถรองรับผู้ใช้งานได้มากกว่าหนึ่งคน
- สามารถใช้งานแอพพลิเคชั่นเพิ่มเติม เช่น แอพพลิเคชั่นที่สามารถวาดรูปได้

1.5 ผลที่คาดว่าจะได้รับ

- ได้ศึกษาและเรียนรู้การทำงานของเทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรด
- สามารถนำความรู้ที่ได้รับไปพัฒนาต่อยอดเพื่อนำไปใช้จริงในด้านอื่นๆได้
- ผู้ใช้มีความพึงพอใจในการใช้ระบบนี้ในการทำงาน
- ลดการใช้เอกสาร ซึ่งมาจากทรัพยากรธรรมชาติที่มีอยู่จำกัด

1.6 อุปกรณ์ที่ต้องใช้

1.6.1 ฮาร์ดแวร์

1) เครื่องฉายภาพ 1 เครื่อง
2) กล้องอินฟราเรด 2 ตัว
3) แผ่นอะคริลิค (Acrylic) 1 แผ่น
4) แผงหลอดแอลอีดี อินฟราเรด 200 หลอด
5) กระดาษไข 1 แผ่น
6) คอมพิวเตอร์ 1 เครื่อง
7) กระจกเงา 2 บาน

1.6.2 ซอฟต์แวร์

1) โปรแกรม Eclip
2) โปรแกรม CCV

1.6.3 ภาษาที่ใช้พัฒนา

1) ภาษา JAVA

1.7 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 ระยะเวลาการดำเนินงาน

Project 1 Project 2

ID	TASK NAME	2013							2014		
		มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1	เสนอหัวข้อโครงงาน										
2	ศึกษาค้นคว้าและทำความเข้าใจ										
3	วิเคราะห์ออกแบบองค์ประกอบระบบและจัดซื้ออุปกรณ์										
4	จัดทำในส่วนของฮาร์ดแวร์										
5	ทดสอบระบบฮาร์ดแวร์										
6	จัดทำส่วนของซอฟต์แวร์ที่ทำหน้าที่ประมวลในการระบุตำแหน่งการสัมผัส										
7	ซอฟต์แวร์ที่สามารถใช้งานร่วมกับ Microsoft Windows 8										
8	จัดทำส่วนของซอฟต์แวร์ในส่วนของแอพพลิเคชั่นเพิ่มเติม										
9	ทดสอบการใช้งานและปรับปรุงแก้ไข										
10	จัดทำปริญญานิพนธ์										

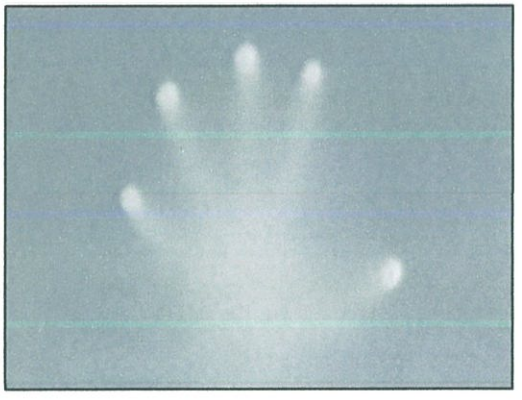
บทที่ 2

ทฤษฎีพื้นฐานที่ใช้

2.1 ส่วนฮาร์ดแวร์

2.1.1 หน้าจอสัมผัสโดยใช้เทคโนโลยีอินฟราเรด

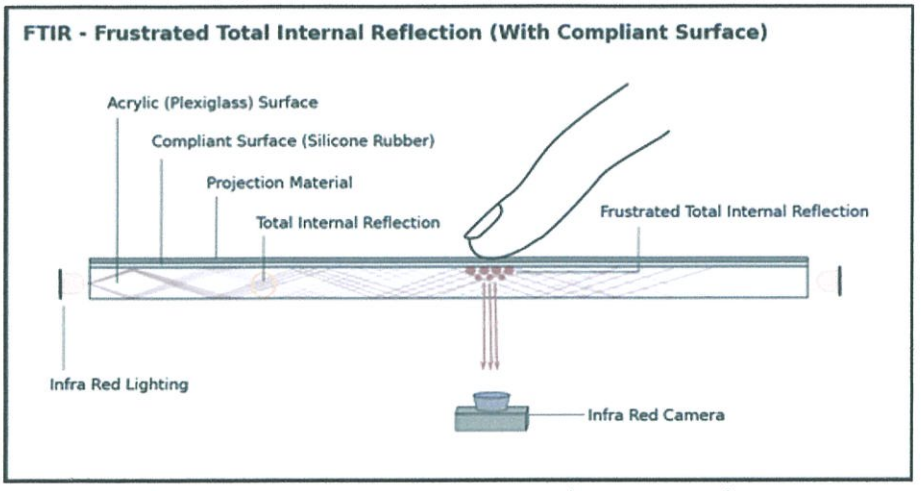
หน้าจอสัมผัสโดยใช้เทคโนโลยีอินฟราเรด เป็นเทคโนโลยีที่ใช้แสงอินฟราเรดเป็นแหล่งกำเนิดแสง เพื่อให้สามารถระบุตำแหน่งของการสัมผัสได้ ซึ่งเป็นเทคโนโลยีที่มีต้นทุนต่ำ และมีความสะดวกในการติดตั้ง ซึ่งเทคโนโลยีอินฟราเรดมี 6 เทคนิคด้วยกัน โดยแต่ละเทคนิคจะใช้กล้องอินฟราเรดเพื่อหาจุดที่สัมผัสได้ ดังรูปที่ 2.1 แต่จะแตกต่างกันในการใช้แหล่งกำเนิดแสงอินฟราเรด ซึ่งอธิบายได้ดังนี้



รูปที่ 2.1 รูปที่ได้จากกล้องอินฟราเรดเมื่อมีการสัมผัส [1]

❖ Frustrated Total Internal Reflection (FTIR)

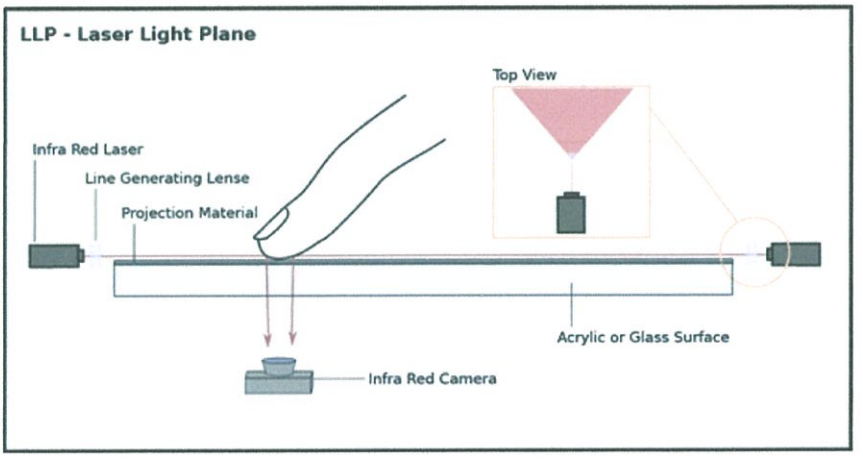
Frustrated Total Internal Reflection เป็นเทคนิคที่ใช้หลักการสะท้อนของแสงภายในวัสดุที่เป็นหน้าจอสัมผัส หลักการคือเมื่อแสงเดินทางภายในวัสดุที่มีค่าดัชนีหักเหของแสงมากๆ จะทำให้มุมหักเหของแสงนั้นมีค่าเท่ากับมุมวิกฤต และหากมีค่าดัชนีหักเหของแสงมากพอก็จะทำให้เกิดการสะท้อนภายในวัสดุนั้น เรียกว่า “Total Internal Reflection” ซึ่งวัสดุที่มีคุณสมบัตินี้คือ อะคริลิค และต้องทำเป็นกรอบแอลอีดี (LED Frame) เพื่อจ่ายแสงเข้าไปในวัสดุ เมื่อมีการสัมผัสบนหน้าจอที่ใช้เทคนิค FTIR จะทำให้เกิดการเปลี่ยนค่าของดัชนีหักเหของแสงทำให้แสงกระจายออกมาตรงจุดที่สัมผัส เมื่อนำกล้องอินฟราเรดมาจับภาพทำให้สามารถระบุตำแหน่งที่สัมผัสหน้าจอได้ แต่หากเกิดจากการสัมผัสจากนิ้วมือที่แห้ง หรือวัสดุที่ทำให้ค่าดัชนีของแสงเปลี่ยนไปไม่มาก จะทำให้ไม่สามารถระบุตำแหน่งได้ เทคนิค FTIR จึงต้องใช้ Compliant Layer ซึ่งส่วนใหญ่ใช้เป็นยางซิลิโคน เพื่อช่วยให้เกิดการกระจายของแสงมากขึ้น ดังรูปที่ 2.2



รูปที่ 2.2 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอ และชิ้นส่วนที่จำเป็นสำหรับเทคนิค FTIR [2]

❖ Laser Light Plane (LLP)

Laser Light Plane ใช้หลักการกำเนิดแสงอินฟราเรดจากเลเซอร์โดยการฉายแสงไปบนพื้นผิวของหน้าจอ เมื่อสัมผัสจะทำให้เกิดการกระจายของแสงลงมาทำให้กล้องอินฟราเรดสามารถระบุตำแหน่งของการสัมผัสได้ ดังรูปที่ 2.3 โดยเทคนิค LLP นี้จะใช้เลเซอร์ ในช่วงของความยาวเคลื่อน 780 นาโนเมตร (nm) และ 940 นาโนเมตร (nm) โดยสิ่งที่ต้องระมัดระวังคือต้องรู้ว่าลำแสงเลเซอร์นั้นเดินทางอย่างไร เนื่องจากเลเซอร์เป็นอันตรายต่อสายตาและช่วงแสงอินฟราเรดเป็นช่วงแสงที่มนุษย์ไม่สามารถมองเห็นได้

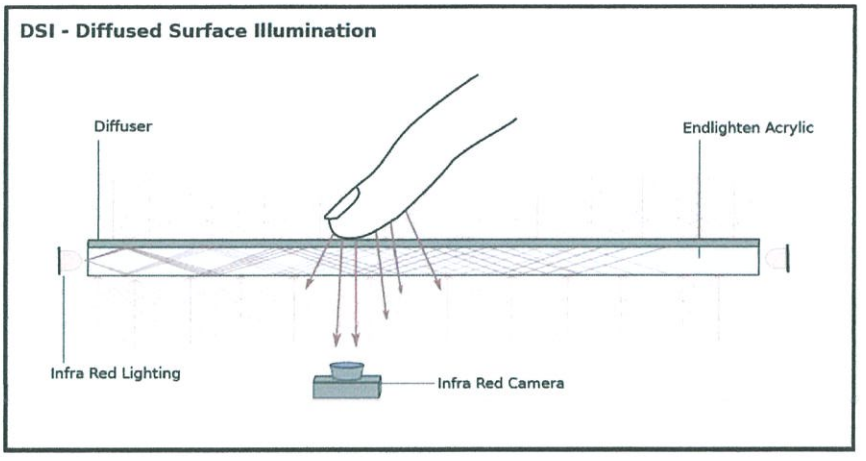


รูปที่ 2.3 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอ และชิ้นส่วนที่จำเป็นสำหรับเทคนิค LLP [3]

❖ Diffused Surface Illumination (DSI)

Diffused Surface Illumination ใช้หลักการคล้ายๆกับเทคนิค FTIR โดยการจ่ายแสงไปรอบๆ อะคริลิค แต่แตกต่างคือแบบ DSI ใช้ อะคริลิคที่มีคุณสมบัติในการกระจายแสง (Enlighten Acrylic) ทำให้ไม่ต้องใช้

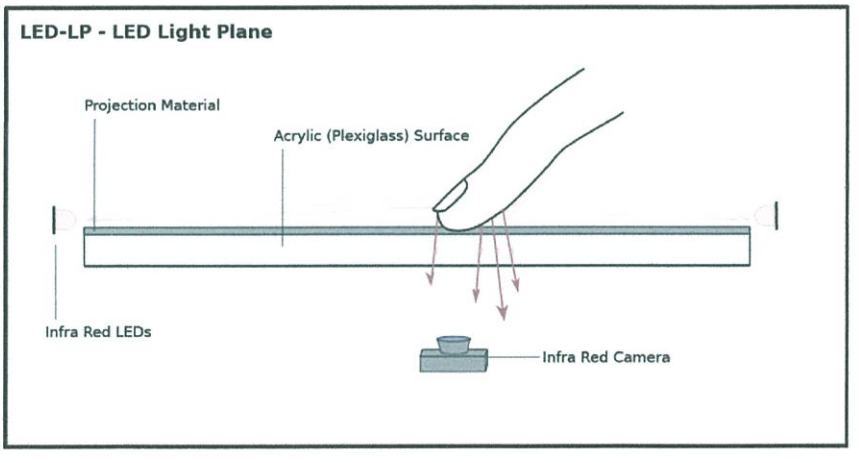
Compliant Layer เพื่อช่วยในการกระจายแสง เมื่อฉายแสงเข้าไปในอะคริลิคชนิดนี้จะกระจายแสงออกเมื่อมีการสัมผัสจะเกิดการหักเหของแสง ทำให้ระบุตำแหน่งของการสัมผัสได้ ดังรูปที่ 2.4



รูปที่ 2.4 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค DSI [4]

❖ LED Light Plane (LED-LP)

LED Light Plane เป็นการใช้แหล่งกำเนิดแสงอินฟราเรดเป็นแบบกรอบแอลอีดี เหมือนเทคนิค FTIR แต่ใช้หลักการฉายแสงด้านบนหน้าจอสัมผัสเหมือนแบบ LLP โดยเมื่อสัมผัสที่หน้าจอสัมผัสจะเกิดการสะท้อน ทำให้สามารถระบุตำแหน่งที่สัมผัสได้เช่นเดียวกับแบบ LLP ดังรูปที่ 2.5

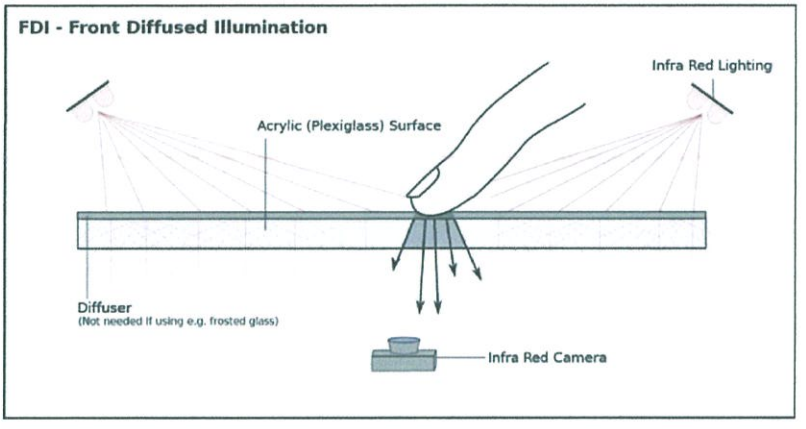


รูปที่ 2.5 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอ
และชิ้นส่วนที่จำเป็นสำหรับเทคนิค LED – LP [5]

❖ Diffused Illumination (DI)

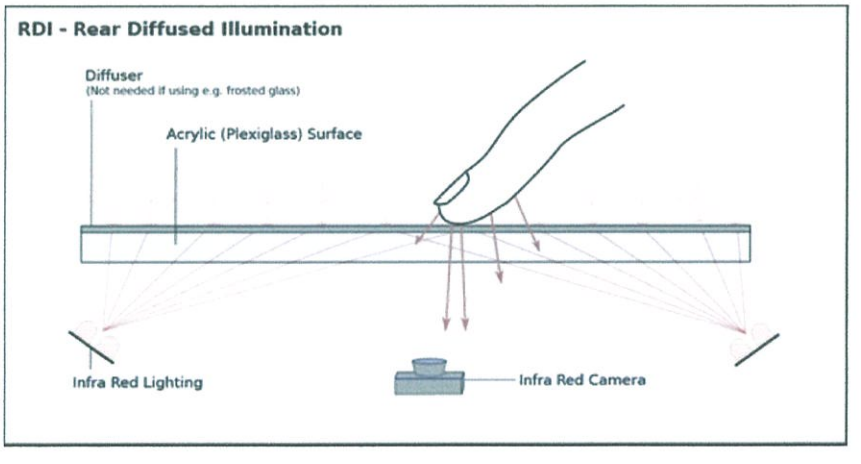
Diffused Illumination มี 2 แบบด้วยกันคือ Front Diffused Illumination และ Rear Diffused Illumination โดยทั้งสองแบบอาศัยหลักการความแตกต่างของพื้นที่ที่มีแสงมากกับพื้นที่ที่ไม่มีแสงหรือมีแสงน้อย (contrast) โดยแสงในที่นี้คือแสงอินฟราเรด ทั้ง 2 เทคนิคมีรูปแบบดังนี้

แบบ Front Diffused Illumination (Front DI) เป็นการให้แสงอินฟราเรดด้านบนของหน้าจอสัมผัสมีมากกว่าอีกด้านของจอสัมผัสโดยใช้โคมไฟขนาดใหญ่ เมื่อสัมผัสจะเกิดความแตกต่างของภาพทำให้สามารถระบุตำแหน่งที่สัมผัสได้ ดังรูปที่ 2.6



รูปที่ 2.6 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค Front DI [6]

แบบ Rear Diffused Illumination (Rear DI) เป็นการให้แสงอินฟราเรดด้านล่างของหน้าจอสัมผัสมีมากกว่าด้านบน เมื่อมีการสัมผัสจะเกิดความแตกต่างของภาพทำให้สามารถระบุตำแหน่งที่สัมผัสได้ ดังรูปที่ 2.7



รูปที่ 2.7 การกระจายของแสงอินฟราเรดเมื่อมีการสัมผัสที่หน้าจอและชิ้นส่วนที่จำเป็นสำหรับเทคนิค Rear DI [7]

จาก 6 เทคนิคของเทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรด ที่กล่าวมาข้างต้นสามารถสรุปข้อดีข้อเสียของเทคนิคต่างๆได้ ดังนี้

- Frustrated Total Internal Reflection (FTIR)

ตารางที่ 2.1 ข้อดีข้อเสียของเทคนิค FTIR

ข้อดี	ข้อเสีย
ไม่ต้องการกล่องปิดเนื่องจากใช้การฉายแสงรอบๆ อะคริลิค	ต้องใช้แหล่งกำเนิดแสงอินฟราเรดแบบกรอบแอลอีดี
สามารถหาจุดสัมผัสได้ง่ายมีการกระจายของแสงชัดเจน	ต้องมี Compliant Surface ซึ่งสามารถทำได้ยาก
สามารถรองรับการสัมผัสได้จากทุกวัตถุ	
สามารถใช้กับการสัมผัสเล็กๆเช่นปากกาได้	

- Rear Diffused Illumination (Rear DI)

ตารางที่ 2.2 ข้อดีข้อเสียของเทคนิค Rear DI

ข้อดี	ข้อเสีย
ไม่จำเป็นต้องใช้ Compliant Surface	ยากที่จะทำให้แสงกระจายอยู่ทั่ว
ไม่จำเป็นต้องใช้ แสงอินฟราเรดแบบกรอบแอลอีดี	ต้องใช้กล่องปิด

- Front Diffused Illumination (Front DI)

ตารางที่ 2.3 ข้อดีข้อเสียของเทคนิค Front DI

ข้อดี	ข้อเสีย
ไม่ต้องใช้กล่องปิด	จุดสัมผัสผิดพลาดได้ง่าย
ติดตั้งได้ง่าย	เกิดการรบกวนจากสิ่งแวดล้อมได้ง่าย ทำให้ไม่มีความน่าเชื่อถือ

- Laser Light Plane (LLP)

ตาราง 2.4 ข้อดีข้อเสียของเทคนิค LLP

ข้อดี	ข้อเสีย
ไม่จำเป็นต้องใช้ Compliant Surface	มีความแม่นยำต่ำ
ไม่ต้องใช้ แสงอินฟราเรดแบบกรอบแอลอีดี	หากสัมผัสจุดที่มีการหักเหของแสงไปแล้วจะไม่

	สามารถระบุตำแหน่งได้
ติดตั้งได้ง่าย	

- Diffused Surface Illumination (DSI)

ตาราง 2.5 ข้อดีข้อเสียของเทคนิค DSI

ข้อดี	ข้อเสีย
ไม่จำเป็นต้องใช้ Compliant Surface	ต้องใช้อะคริลิคชนิดพิเศษ (Enlighten Acrylic) ซึ่งมีราคาสูง
สามารถปรับเปลี่ยนไปใช้แบบ FTIR ได้	ความแตกต่างของจุดสัมผัสต่ำกว่าแบบ FTIR

- LED Light Plane (LED-LP)

ตาราง 2.6 ข้อดีข้อเสียของเทคนิค LED-LP

ข้อดี	ข้อเสีย
ไม่จำเป็นต้องใช้ Compliant Surface	ลำแสงแคบมาสามารถใช้กับหน้าจอขนาดใหญ่ได้
ไม่ต้องใช้กล่องปิด	ความแม่นยำต่ำ
	หากสัมผัสจุดที่มีการหักเหของแสงไปแล้วจะไม่สามารถระบุตำแหน่งได้

2.1.2 กล้องเว็บแคม (Web cameras) อินฟราเรด

จากเทคโนโลยีหน้าจอสัมผัสอินฟราเรดนั้นต้องใช้ กล้องอินฟราเรดในการรับภาพแต่เนื่องจาก กล้องอินฟราเรดนั้นมีต้นทุนที่สูง จึงนำกล้องเว็บแคมที่ใช้งานกันทั่วไปมา ปรับเปลี่ยนให้เป็นกล้องอินฟราเรดได้

โดยปกติกล้องเว็บแคมทั่วไปนั้นจะมีการกำจัดแสงอินฟราเรดโดยมีฟิลเตอร์ (Filter) กรองแสงอินฟราเรดออก จึงต้องปรับเปลี่ยนให้ไปในทางตรงกันข้าม โดยนำฟิลเตอร์กรองอินฟราเรดนี้ออก และใส่ฟิลเตอร์ที่กรองแสงในช่วงของแสงอินฟราเรดแทนที่ แต่หากกล้องเว็บแคมที่ใช้มีราคาแพงก็ไม่สามารถนำมาปรับเปลี่ยนได้เนื่องจากตัวกรองอินฟราเรดจะติดไปกับตัวเลนส์ของกล้อง

คุณสมบัติหลักของกล้องที่ใช้ได้กับหน้าจอสัมผัสแบบอินฟราเรดมีดังนี้

- ความละเอียดของภาพ

  ความละเอียดของภาพมีผลเป็นอย่างมากในการระบุตำแหน่งสัมผัสสำหรับพื้นที่ผิวสัมผัสขนาดเล็กสามารถใช้ความละเอียดภาพตั้งแต่ 320 x 240 pixels และสำหรับพื้นที่ขนาดใหญ่ความมีความละเอียดตั้งแต่ 640 x 480 pixels ขึ้นไป
- อัตราการรับส่งภาพ

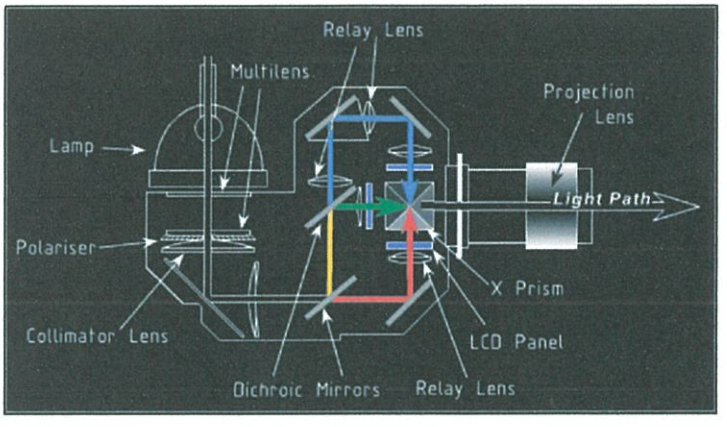
  อัตราการส่งภาพ คือจำนวนภาพที่ส่งใน 1 วินาที ซึ่งมีผลต่อการเคลื่อนที่ของการสัมผัส กล้องที่นำมาใช้ควรมีอัตราการส่งภาพอย่างน้อย 30 เฟรมต่อวินาที (Frames per second : FPS)
- ระยะการจับภาพของกล้อง (Capture size)

  ระยะการจับภาพของกล้อง ต้องทำให้สามารถจับได้ทุกส่วนของการสัมผัส

2.1.3 การแสดงผลของหน้าจอสัมผัสแบบอินฟราเรด โดยใช้เครื่องฉายภาพ

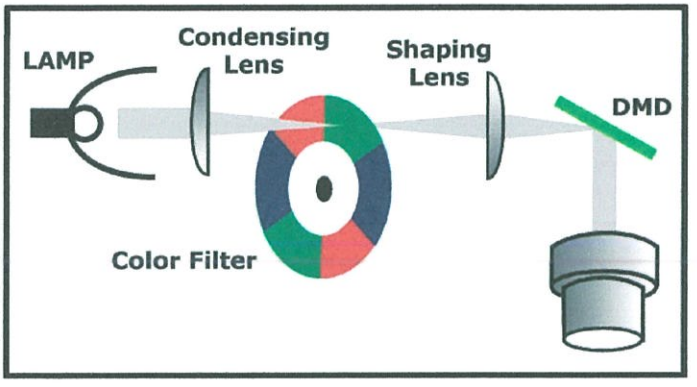
การใช้เครื่องฉายภาพเป็นการแสดงผลที่นิยมในหน้าจอขนาดใหญ่ โดยทั่วไปเครื่องฉายภาพแบ่งออกเป็น 2 แบบ เลือกใช้ตามรูปแบบของการใช้งาน

เครื่องฉายภาพแบบ LCD (Liquid Crystal Display) เป็นเทคโนโลยีของเครื่องฉายภาพที่มีการบรรจุแผ่น LCD จำนวน 3 แผ่นไว้ในตัวเครื่องเพื่อใช้ในการสร้างภาพ ซึ่งเปรียบเสมือนแม่สี 3 สี โดยจะมีสีแดง เขียว และน้ำเงิน โดยแสงจะวิ่งผ่านพาเนล LCD นี้ ซึ่งจะเป็นตัวที่เปิดหรือปิดการทำงานของพิกเซลในแต่ละเม็ดสี และเมื่อสร้างภาพเป็นที่เรียบร้อยแล้วก็จะส่งออกไปยังเครื่องและฉายต่อไป ดังรูปที่ 2.8



รูปที่ 2.8 หลักการทำงานของเครื่องฉายภาพแบบ LCD [8]

เครื่องฉายภาพแบบ DLP (Digital Light Processing) ใช้หลักการที่แสงวิ่งผ่านจานแม่สี โดยแสงที่วิ่งผ่านจานแม่สีแดง เขียว และน้ำเงินโดยแต่ละสีจะวิ่งผ่านไปที่กระจกขนาดเล็กในแผ่น DMD โดยถ้าเกิดกระแสไฟฟ้า กระจกขนาดเล็กก็จะสะท้อนแสงแม่สีไฟที่จอภาพ เพื่อที่จะผสมสีให้เกิดภาพ โดยจะมีวงจรควบคุมให้แผ่นจานสีหมุนให้ได้จังหวะกับการผลิดเอียงมุมของกระจกขนาดเล็กในแผ่น DMD ดังรูปที่ 2.9 เครื่องฉายภาพแบบ DLP ให้ค่าความสว่างสูง , ความคมชัดที่สูง , ค่า Black level ที่ดีทำให้สีดำสนิท และการเกลี่ยแสง ที่ดีเหมาะสำหรับงาน ที่ต้องการความคมชัดมาก



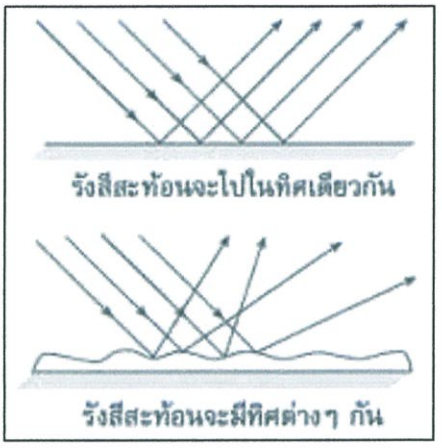
รูปที่ 2.9 หลักการทำงานของเครื่องฉายภาพแบบ DLP [8]

ตารางที่ 2.7 เปรียบเทียบข้อดีข้อเสียของเครื่องฉายภาพแบบ LCD และ แบบ DLP

เครื่องฉายภาพแบบ	ข้อดี	ข้อเสีย
LCD (Liquid Crystal Display)	ให้กำลังแสงได้ดี มีความคมชัดของสีที่ชัด	ขนาดใหญ่เกิดปัญหาในเรื่องของ Dead Pixels ทั้งแบบถาวรและไม่ถาวรได้ และจะส่งผลต่อการแสดงภาพ ตัวจอ LCD สามารถเสียหายได้และมีราคาที่แพงมากกว่า DLP
DLP (Digital Light Processing)	มีจุดพิกเซลที่ละเอียด อายุในการใช้งานก็ยาวนานกว่า	ความอิ่มหรือความชุ่มฉ่ำของสีจะด้อยกว่า LCD

2.1.4 กฎการสะท้อน

การสะท้อนแสง หมายถึง การที่แสงไปกระทบกับตัวกลางแล้วสะท้อนไปในทิศทางอื่นหรือสะท้อนกลับมาทิศทางเดิม การสะท้อนของแสงนั้นขึ้นอยู่กับพื้นผิวของวัตถุด้วยว่าเรียบหรือหยาบโดย ทั่วไปพื้นผิวที่เรียบและมันจะทำให้มุมของแสงที่ตกกระทบมีค่าเท่ากับมุม สะท้อนตำแหน่งที่แสงตกกระทบกับแสงสะท้อนบนพื้นผิวจะเป็นตำแหน่งเดียวกัน ดังรูปที่ 2.10

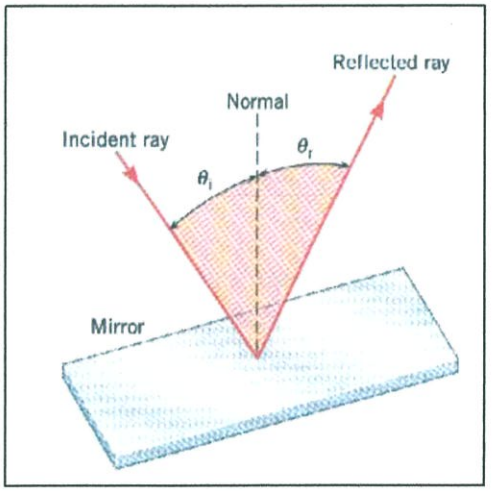


รูปที่ 2.10 การสะท้อนแสงของพื้นที่ผิวเรียบและขรุขระ [9]

- มุมตกกระทบคือมุมที่รังสีตกกระทบ (Incident ray) ทำกับเส้นปกติ (Normal) ของผิวสะท้อน
- มุมสะท้อน (Reflected ray) คือมุมที่รังสีสะท้อนทำกับเส้นปกติ

การสะท้อนของแสงที่มีระเบียบจะได้

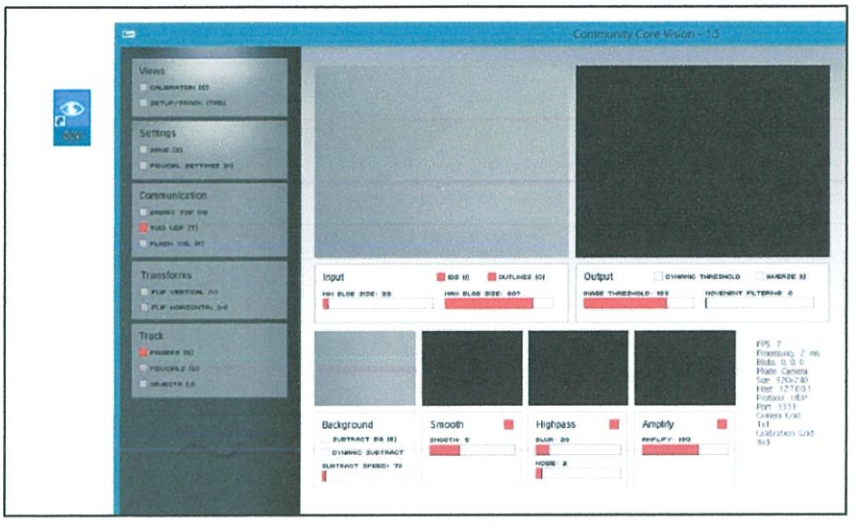
1) มุมตกกระทบมีค่าเท่ากับมุมสะท้อน
2) รังสีตกกระทบ รังสีสะท้อนและเส้นปกติ จะอยู่ในระนาบเดียวกัน



รูปที่ 2.11 การสะท้อนของกระจกเงา[10]

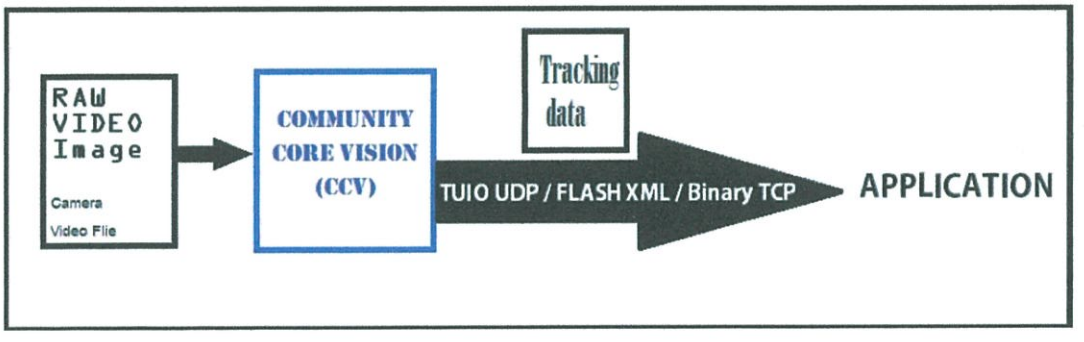
2.2 ส่วนซอฟต์แวร์

2.2.1 Community Core Vision (CCV)



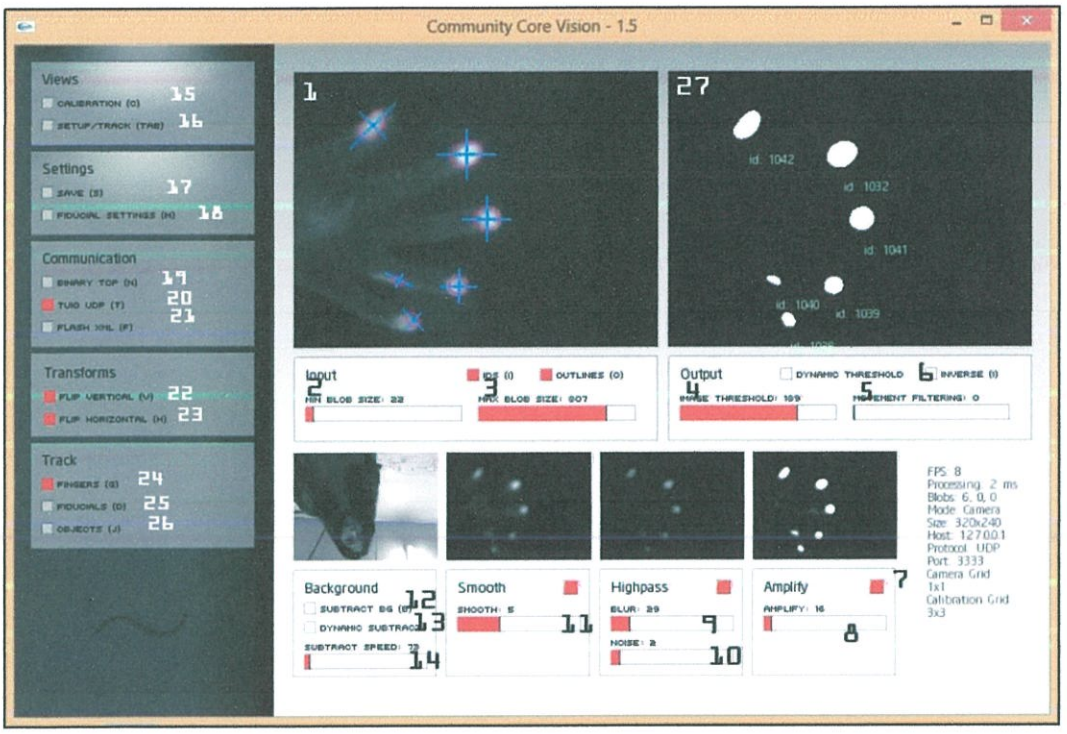
รูปที่ 2.12 โปรแกรม Community Core Vision (CCV)

โปรแกรม Community Core Vision (CCV) เป็นโปรแกรมที่มีการรับอินพุตเป็นวีดีโอ และเอาต์พุตเป็นข้อมูลของการระบุตำแหน่ง (เช่น ไอดีระบุตำแหน่งของการสัมผัส ขนาดของจุดสัมผัส (Blob size)) ซึ่งเป็นโปรแกรมที่ใช้ในการสร้างแอพพลิเคชั่นของหน้าจอสัมผัสหลายจุด (Multi-touch Application) โปรแกรม CCV สามารถเชื่อมต่อได้หลากหลายโปรโตคอล TUIO / Binary TCP / XML และรองรับเทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรดได้ ดังรูปที่ 2.13



รูปที่ 2.13 หลักการเบื้องต้นของโปรแกรม CCV

การใช้งานโปรแกรม Community Core Vision (CCV)



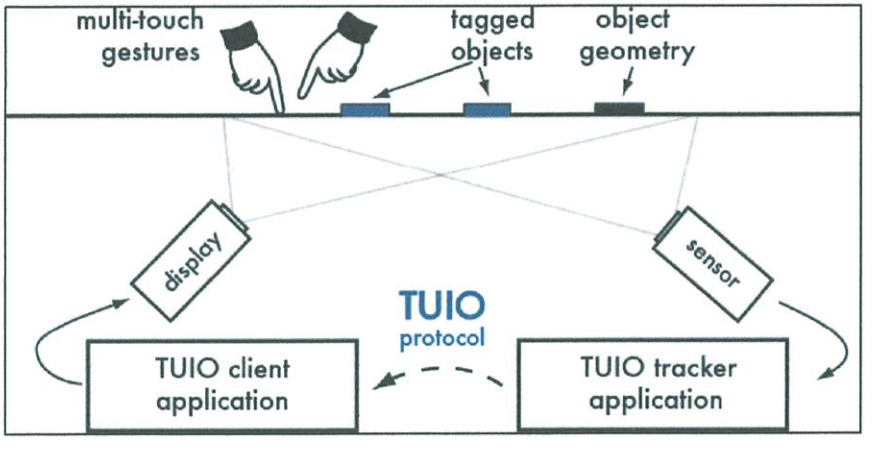
รูปที่ 2.14 การใช้งานโปรแกรม Community Core Vision (CCV)

1. Source image – แสดงวีดีโอจากกล้องหรือไฟล์วีดีโอ
2. Min Blob Size – กำหนดจุดสัมผัสที่เล็กที่สุดที่ต้องการ หากมีจุดที่เล็กกว่าที่กำหนดจะไม่ทำการระบุตำแหน่ง (ไม่มีไอดี)
3. Max Blob Size - กำหนดจุดสัมผัสที่ใหญ่ที่สุดที่ต้องการ หากมีจุดที่ใหญ่กว่าที่กำหนดจะไม่ทำการระบุตำแหน่ง (ไม่มีไอดี)
4. Threshold Slider – ปรับระดับของพิกเซลของการระบุตำแหน่งที่สามารถยอมรับได้
5. Movement filtering – ปรับระดับของระยะทาง ก่อนที่จะมีการตรวจพบจุดสัมผัส
6. Inverse – เปลี่ยนจุดสัมผัสสีดำเป็นสีขาว
7. On/off Toggle – เปิด/ปิด การใช้งานตัวกรองต่าง
8. Amplify Slider – ทำให้จุดตำแหน่งสัมผัสที่อ่อน ชัดขึ้น
9. Highpass Blur Slider – นำส่วนที่เบลอออก และนำส่วนที่ชัดให้มีความคมชัดขึ้น
10. Highpass Noise – กรองในส่วนที่เป็น noise ออก
11. Smooth Slider – ปรับความเรียบเนียนของภาพ
12. Remove Background Button – การเก็บภาพที่เป็นพื้นหลังไว้ และนำมาลบออกเมื่อได้ภาพใหม่เพื่อเป็นการลบพื้นหลังที่รบกวนระบบออก

13. Dynamic Subtract Toggle – กำหนดการเปลี่ยนแปลงของสิ่งแวดล้อมที่เป็นพื้นหลัง เพื่อไม่ให้รบกวนระบบ
14. Subtract Speed – ปรับค่าการเปลี่ยนแปลงของพื้นหลังเพื่อไม่ให้รบกวนระบบ
15. Calibration – การปรับตั้งค่าหน้าจอให้ตรงกับหน้าจอที่ใช้งานจริง
16. Setup/Track – การกำหนดอินพุต และการตั้งค่ากล้องที่ใช้
17. Save – บันทึกค่า
18. Fiducial Setting – การตั้งค่าความแม่นยำ
19. Communication by Binary TCP – การส่งข้อมูลโดยใช้ Binary TCP
20. Communication by TUIO UDP - การส่งข้อมูลโดยใช้ TUIO UDP
21. Communication by FLASH XML – การส่งข้อมูลโดยใช้ FLASH XML
22. Flip Vertical – สลับด้านแนวตั้ง
23. Flip Horizontal – สลับด้านแนวนอน
24. Track Fingers – ทำการระบุตำแหน่งนิ้วมือ
25. Track Fiducials – ทำการระบุตำแหน่งที่ต้องการความแม่นยำ
26. Track Objects – ทำการระบุตำแหน่งวัตถุทั่วไป

2.2.2 โปรโตคอล TUIO

TUIO เป็นโปรโตคอลพื้นฐานที่ใช้งานกันทั่วไป และเป็น API (Application Program Interface: ทำหน้าที่เชื่อมต่อระหว่างแอพพลิเคชั่นกับระบบปฏิบัติการ) TUIO ทำหน้าที่ส่งรายละเอียดของการระบุตำแหน่ง รวมทั้งการสัมผัสต่างๆจากนิ้วมือหรือวัตถุ ไปสู่แอพพลิเคชั่น โดยจะทำการเข้ารหัสควบคุมข้อมูลจากโปรแกรมระบุตำแหน่ง (tracker application) และส่งไปยังแอพพลิเคชั่นต่างๆที่มีความสามารถในการถอดรหัสโปรโตคอลนี้ ดังรูปที่ 2.15



รูปที่ 2.15 หลักการทำงานของโปรโตคอล TUIO [11]

- การทำงานของโปรโตคอล TUIO

  การเข้ารหัส : Open sound control

  การส่งข้อมูล : UDP (User Datagram Protocol)

  พอร์ต: 3333

- มี 4 message classes

  1) SET : ใช้ติดต่อข้อมูลระหว่างวัตถุ เช่น ตำแหน่ง การปรับ

  2) ALIVE : ใช้เก็บ session ids ของวัตถุที่แสดงผลอยู่

  3) FSEQ : ใช้เป็น tag เฉพาะที่เก็บการเปลี่ยนแปลงของ frame id

  4) SOURCE : ระบุ TUIO source เพื่อที่จะ multiplex source บนฝั่งของ client ได้ (message เสริม)

- สรุปการทำงานของ message

  1) attribute ของ object จะถูกส่งหลังจากที่ state เปลี่ยนโดยใช้ SET message

  2) client รู้ object ว่ามี object เพิ่มหรือลดจาก SET หรือ ALIVE message

  3) object จะถูกลบเมื่อได้รับ ALIVE message

  4) FSEQ message จะถูกรวมไว้ใน bundle

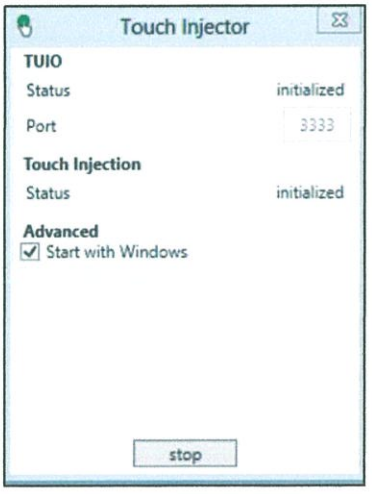
  5) SOURCE message จะระบุ application และ address

- การทำงานของแอพพลิเคชั่นแบ่งได้ 2 ส่วนด้วยกันคือ

  1) ส่วนที่รับค่ามาจากเซ็นเซอร์ (TUIO tracker application)

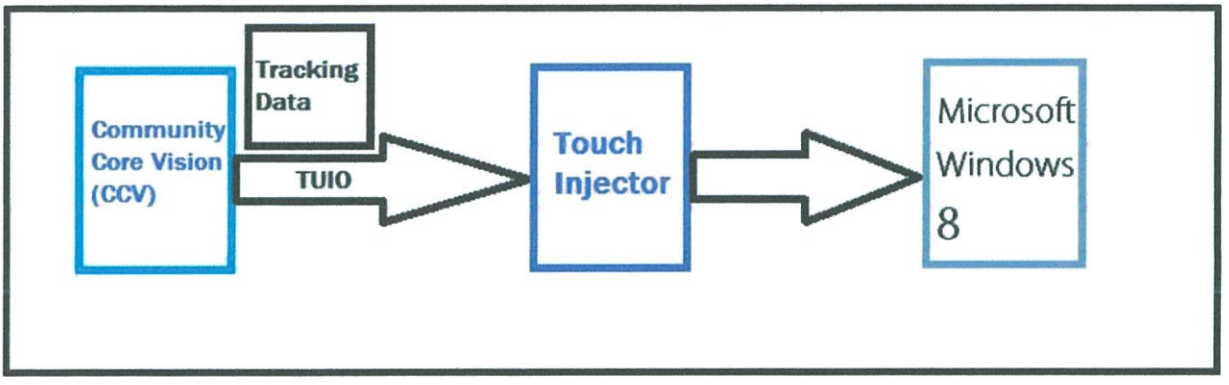
  2) ส่วนที่แสดงผลออกทางหน้าจอ (TUIO client application)

2.2.3 Touch Injector



รูปที่ 2.16 โปรแกรม Touch Injector

Touch Injector เป็นโปรแกรมสำหรับ Microsoft Windows 8 ที่มีหน้าที่ทำการสร้างการสัมผัสหน้าจอที่ได้รับมาจากโปรโตคอล TUIO อาจเรียกได้ว่าเป็น touch driver ถึงแม้ว่าจะไม่ใช่ driver เมื่อโปรแกรม CCV (Community Core Vision) ส่งข้อมูลออกมาผ่าน TUIO เป็นตำแหน่งของนิ้วโปรแกรม Touch Injector จะนำข้อมูลนั้นมาส่งต่อให้ Windows 8 ดังรูปที่ 2.17



รูปที่ 2.17 หลักการทำงานของ Touch Injector

2.2.4 ภาษาจาวา

รูปที่ 2.18 จาวา

ภาษาจาวา (JAVA) เป็นภาษาที่ใช้ในการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) และให้สามารถทำงานบนเครื่องคอมพิวเตอร์ที่มีสภาพแวดล้อมต่างกันโดยไม่ต้องทำการคอมไพล์ใหม่ซึ่งเรียกคุณสมบัตินี้ว่า Platform Independent ซึ่งจุดเด่นของภาษาจาวามีดังนี้

- ชุดพัฒนาโปรแกรมสามารถใช้งานได้ฟรี
- เป็นภาษาที่ออกแบบมาให้ใช้งานง่าย สามารถใช้งานโดยนำพื้นฐานของภาษา C มาใช้และตัดความยุ่งยากของภาษา C เช่นพอยเตอร์
- สามารถทำงานบนเครื่องคอมพิวเตอร์ที่ต่างระบบกันได้

- สนันสนุนการเขียนโปรแกรมเชิงวัตถุ
- ทำงานบนเว็ปบราวเซอร์ได้
- มีคลาสต่างๆใช้เรียกใช้จำนวนมาก
- สามารถคำนวณพื้นที่หน่วยความจำได้อัตโนมัติและภาษาจาวาสามารถคืนหน่วยความจำให้กับออบเจ็กต์ที่ไม่มีการใช้โดยอัตโนมัติ

ไลรารี่ที่ใช้กับการเขียนแอพพลิเคชั่น

ไลบรารี่ที่ใช้กับการเขียนแอพพลิเคชั่นเป็นไลบราลี่ของเว็ปไซต์ MT4j – Multitouch for JAVA

AbstractMTApplication : ใช้สำหรับการสร้างมัลติทัชแอพพลิเคชั่น
components.TransformSpace: เป็นการดึงค่าพื้นที่ออกมาซึ่งเป็นค่าที่ได้ประกาศ (Enum Tranformspace)
components.visibleComponents.shapes.MTRoundRectangle: สร้างหน้าต่างสี่เหลี่ยม
components.visibleComponents.shapes.MTRectangle.PositionAnchor: หาตำแหน่งบนหน้าต่างสี่เหลี่ยม
components.visibleComponents.widgets.MTBackgroundImage: ใช้ภาพพื้นหลัง
components.visibleComponents.widgets.MTList: สร้างลิส (List)
components.visibleComponents.widgets.MTListCell: สร้างเซลล์ให้กับลิส
components.visibleComponents.widgets.MTSceneTexture: รูปแบบของฉากที่กำหนดขึ้นมา (สามารถกำหนดตำแหน่ง,ความสูงและความกว้าง)
components.visibleComponents.widgets.MTHandler: ทำการสร้างรูปเมื่อคลิกและทำการลบรูปเมื่อกดค้าง (เขียนขึ้นมาเองเพิ่มเติม)
components.visibleComponents.widgets.MTTextArea: กำหนดพื้นที่หลังข้อความ ถ้าไม่มีการกำหนดพื้นที่ พื้นที่ที่แสดงจะขยายตามขนาดของข้อความ แต่หากมีการกำหนดพื้นที่จะทำการตัดคำเพื่อให้ได้ตามพื้นที่ที่กำหนด
input.inputProcessors: การรับอินพุตต่างๆ เช่น การกด การกดค้าง
sceneManagement: การจัดลำดับต่างๆก่อนที่จะทำขั้นต่อไป
util.MTColor: การเติมสี
util.font: ประกาศใช้ตัวอักษรและรูปแบบตัวอักษร
util.math.Vector3D: กำหนดตำแหน่งโดยใช้หลักแกน x y z
processing.core.PImage : โหลดข้อมูลที่เป็นรูปภาพ

ภาษาจาวาเบื้องต้น

```
import org.mt4j.AbstractMTApplication;  0
                                        1
public class myTouchScript extends AbstractScene {

 2  public myTouchScript(AbstractMTApplication mtApplication, String name) {
        |
     3  super(mtApplication, name);
     4  this.setClearColor(new MTColor(255, 128, 0, 255));
        this.registerGlobalInputProcessor(new CursorTracer(mtApplication, this));
        MTTextArea textArea = new MTTextArea(mtApplication, FontManager.getInstance().createFont(
                mtApplication,
                "arial.ttf", //Font name
     5          50, //Font size
                new MTColor(255, 255, 255, 255), //Font fill color
                new MTColor(255, 255, 255, 255))); //Font outline color
     6  textArea.setNoFill(true);
     7  textArea.setNoStroke(true);
     8  textArea.setText("KMITL");
     9  this.getCanvas().addChild(textArea);
        textArea.setPositionGlobal(new Vector3D(mtApplication.width/2f, mtApplication.height/2f));

    }

    @Override
 10 public void init() {

    }

 11 @Override
    public void shutDown() {

    }

}
```

รูปที่ 2.19 ภาษาจาวาเบื้องต้น (1)

0 คือ การ import เพื่อใช้สร้าง multitouch application ตัวใหม่

1 คือ การสร้าง class myTouchScript โดยเพิ่มในส่วนของ scene ขึ้นมาเพราะทุกๆ application จำเป็นที่จะต้องมี scene เป็น background

2 คือ การสร้าง constructor โดยมีพารามิเตอร์ mtApplication และ name โดย name สามารถใช้เป็นตัวระบุและรับ scene ได้

3 คือ ใช้สำหรับในการทำงานของ subclass โดย subclass จะเป็นตัวเรียกใช้ซึ่งมีความสามารถดังนี้คือ subclass สามารถใช้ constructor ของ superclass ได้

4 คือ การสร้างสีให้กับ screne ในที่นี้เลือกสีส้ม

5 คือ การสร้างสีและลักษณะของ font ที่จะใช้ภายใน screne ในที่นี้เลือกอักษรสีขาว, font: arial, ขนาด 50

6, 7 คือ การลบกรอบของ text ที่แสดงภายใน screne เพื่อให้เห็นแต่ตัวอักษรเท่านั้น

8 คือ การแสดงข้อความลงบน screne ในที่นี้เลือกที่จะแสดงคำว่า KMITL

9 คือ การ add text ที่เราพิมพ์ลงไปใน screne โดยจัดให้อยู่กึ่งกลางของ screne

10 คือ ในกรณีที่เราต้องการเขียนตอนเริ่มการทำงานว่าจะให้ทำอะไรได้บ้างสามารถเขียนได้ในนี้

11 คือ ในกรณีที่เราต้องการเขียนตอนปิดการทำงานว่าจะให้ทำอะไรได้บ้างสามารถเขียนได้ในนี้

```
package myTouch;

import org.mt4j.MTApplication;12

                                13
public class myTouchApp extends MTApplication {

    public static void main(String[] args)  {
        initialize(); 14
    }

    @Override      15
    public void startUp() {
        addScene(new myTouchScript(this, "Test my first application")); 16
    }

}
```

รูปที่ 2.20 ภาษาจาวาเบื้องต้น (2)

12 คือ การ import เพื่อใช้สร้าง multitouch application ตัวใหม่

13 คือ การเพิ่มส่วนของการแสดงผล

14 คือ การ กำหนดค่าเริ่มต้นให้กับ application

15 คือ การสร้าง function ในการเริ่มต้นทำการทำงาน

16 คือ การเรียกใช้ object จากคลาสที่สร้างไว้

บทที่ 3

การออกแบบ

3.1 อุปกรณ์ฮาร์ดแวร์

3.1.1 กล้องเว็บแคม (Web cameras) อินฟราเรด

รูปที่ 3.1 กล้อง Logitech C170 [13]

ในปริญญานิพนธ์นี้ใช้กล้อง Logitech C170 (แสดงดังรูปที่ 3.1) ซึ่งเป็นกล้องเว็บแคมที่มีคุณสมบัติที่สามารถใช้กับหน้าจอสัมผัสแบบอินฟราเรดได้และมีราคาที่ไม่สูงโดย กล้อง Logitech C107 มีคุณสมบัติดังนี้

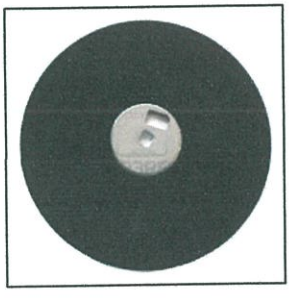
USB Type: 2.0

Frame rate: 640x480 30 FPS

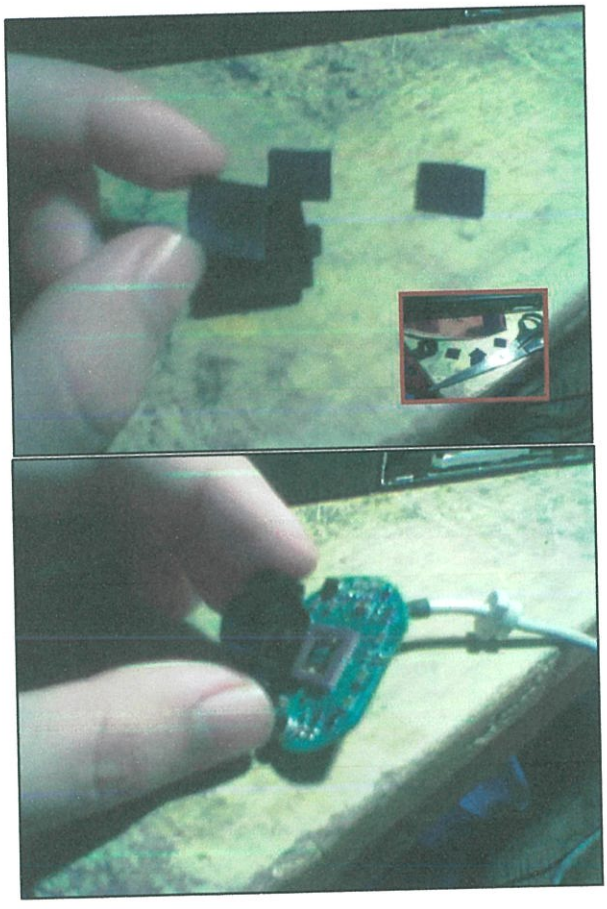
Diagonal Field of View (FOV): 58°

Video Capture (4:3 SD): 320x240, 640x480, 1024x768

เมื่อนำกล้องเว็บแคมมานำส่วนที่เป็นฟิลเตอร์กรองแสงอินฟราเรดออก และใส่ฟิลเตอร์ที่กรองเฉพาะแสงอินฟราเรด (IR-pass filter) เข้าไปแทนที่ โดยในที่นี้ใช้แผ่น Floppy Disk ตัดเป็นชิ้นพอดีกับเลนส์แล้วใส่เข้าไปแทนที่ฟิลเตอร์เดิมที่มีอยู่ ดังรูปที่ 3.3

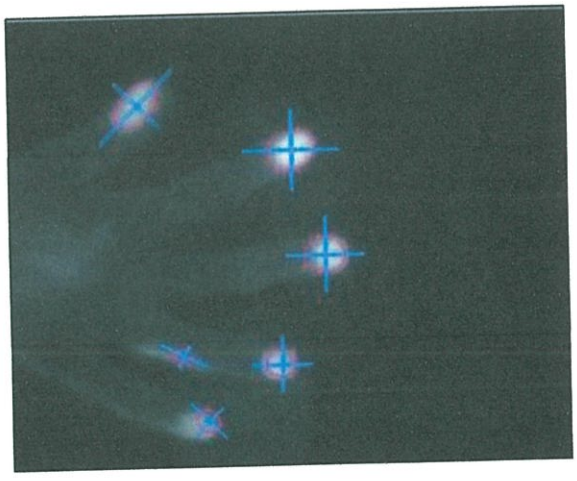


รูปที่ 3.2 แผ่น Floppy Disk



รูปที่ 3.3 การนำส่วนที่เป็นฟิลเตอร์กรองแสงอินฟราเรดออก
และใส่ฟิลเตอร์ที่กรองเฉพาะแสงอินฟราเรด (IR-pass filter) เข้าไปแทนที่ [14]

เมื่อนำกล้องไปรับภาพจากเทคโนโลยีหน้าจอสัมผัสเทคนิคต่างๆก็จะทำให้สามารถเห็นตำแหน่งที่สัมผัสได้ดังรูปที่ 3.4



รูปที่ 3.4 ภาพที่เกิดจากการรับภาพจากเทคโนโลยีหน้าจอสัมผัสอินฟราเรด
เทคนิค Rear DI

3.1.2 การแสดงผลของหน้าจอสัมผัสแบบอินฟราเรด โดยใช้เครื่องฉายภาพ

เครื่องฉายภาพที่ใช้คือ EPSON BusinessEB - S02 (แสดงดังรูปที่ 3.5)ซึ่งสามารถใช้ได้กับเทคโนโลยีหน้าจอสัมผัสอินฟราเรด โดยเครื่องฉายภาพ EPSON BusinessEB - S02 มีคุณสมบัติดังนี้

- ความสว่างสูง 2600 ANSI lumens (standard) และ 2080 ANSI Lumens (Eco)
- เทคโนโลยีการแสดงภาพ แบบ LCD- Poly-silicon TFT Active Matrix ขนาด 0.55 นิ้วx3
- ความละเอียดภาพแบบ SVGA (800x600 pixels)
- อัตราเปรียบต่างของสีขาวและดำ (Contrast Ratio) สูงถึง 3000: 1
- หลอดภาพ แบบ E-TORL 200W แบบ UHE

รูปที่ 3.5 เครื่องฉายภาพ EPSON BusinessEB - S02 [15]

3.1.3 หลอดเอลอีดีอินฟราเรด (Infrared LED)

หลอดแอลอีดีอินฟราเรดที่ใช้คือรุ่น TOIR -50b94 bdEa ซึ่งมีคุณสมบัติดังรูปที่ 3.6

OASIS

TAIWAN OASIS LED DATA SHEET (FOR INFRARED)

PART NO. : TOIR-50b94bCEa

ABSOLUTE MAXIMUM RATINGS AT TA=25°C

PARAMETER	SYMBOL	DATA	UNIT
Forward Current	I_{FM}	100	mA
Peak Forward Current (duty=1:100, f=100KHZ)	I_{FP}	1000	mA
Reverse Voltage	V_R	6	V
Power Dissipation	P_D	150	mW
Operating Temperature Range		−25 to +85	°C
Storage Temperature Range		−30 to +85	°C
Lead Sold Temperature (1/10 Inch Below Seating Plane)		260°C for 3 sec.	

ELECTRICAL/OPTICAL CHARACTERISTICS AT TA=25°C

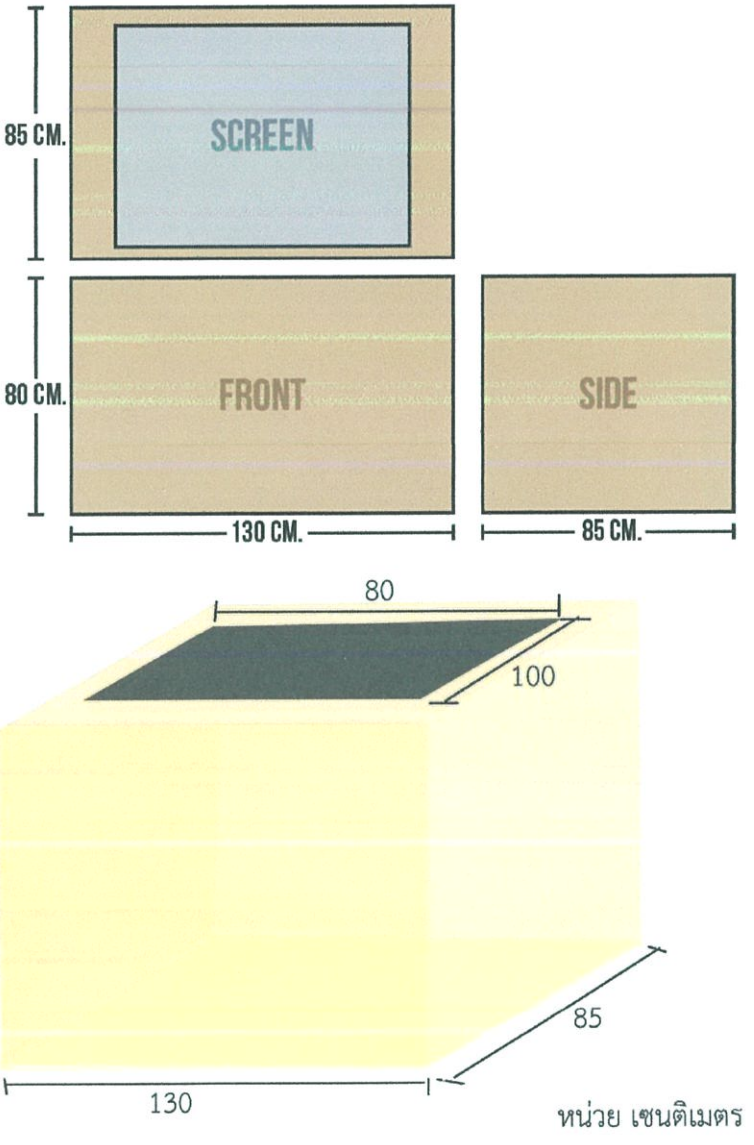
PARAMETER	SYMBOL	DATA	UNIT	TEST CONDITION
Radiated Output Power	$P_{O(Typ.)}$	12.0	mW	Distance: 10cm I_F=50mA Detector Area: 1cm²
Forward Voltage	V_F	TYP: 1.25 MAX: 1.45	V	I_F=20mA
Wavelength	λP	940	nm	I_F=20mA
Spectrum Width of Half Value	$\Delta\lambda$	50	nm	I_F=20mA
Reverse Current	I_R	10	μA	V_R=5V
Full Viewing Angle	$2X\frac{1}{2}\theta$	25	°	I_F=20mA
Lens		Water Clear		
Radiation Material		GaAs/GaAs		

รูปที่ 3.6 คุณสมบัติของหลอดแอลอีดีอินฟราเรดรุ่น TOIR -50b94 bdEa [16]

3.2 การออกแบบฮาร์ดแวร์

3.2.1 โครงสร้างของโต๊ะอัจฉริยะ

ขนาดของโต๊ะออกแบบให้เป็นโต๊ะที่มีขนาดกลางสามารถนำไปใช้งานได้หลากหลายโดยการออกแบบขนาดของหน้าจอคำนึงถึงการฉายภาพของเครื่องฉายภาพซึ่งเครื่องฉายภาพ EPSON BusinessEB - S02 จะฉายภาพได้มีประสิทธิภาพในอัตราส่วน 4:3 ได้ดีกว่าการฉายภาพในอัตราส่วน 16:9 จึงออกแบบขนาดของหน้าจอเป็นขนาด 100 x 75 เซนติเมตรซึ่งหน้าจอนี้ได้ใช้อะคริลิคเนื่องจากมีคุณสมบัติในการหักเหแสงเช่นเดียวกับกระจกแต่มีราคาที่ต่ำกว่าโดยการออกแบบขนาดของส่วนต่างๆของโครงสร้างของโต๊ะอัจฉริยะสามารถแสดงได้ดังรูปที่ 3.7

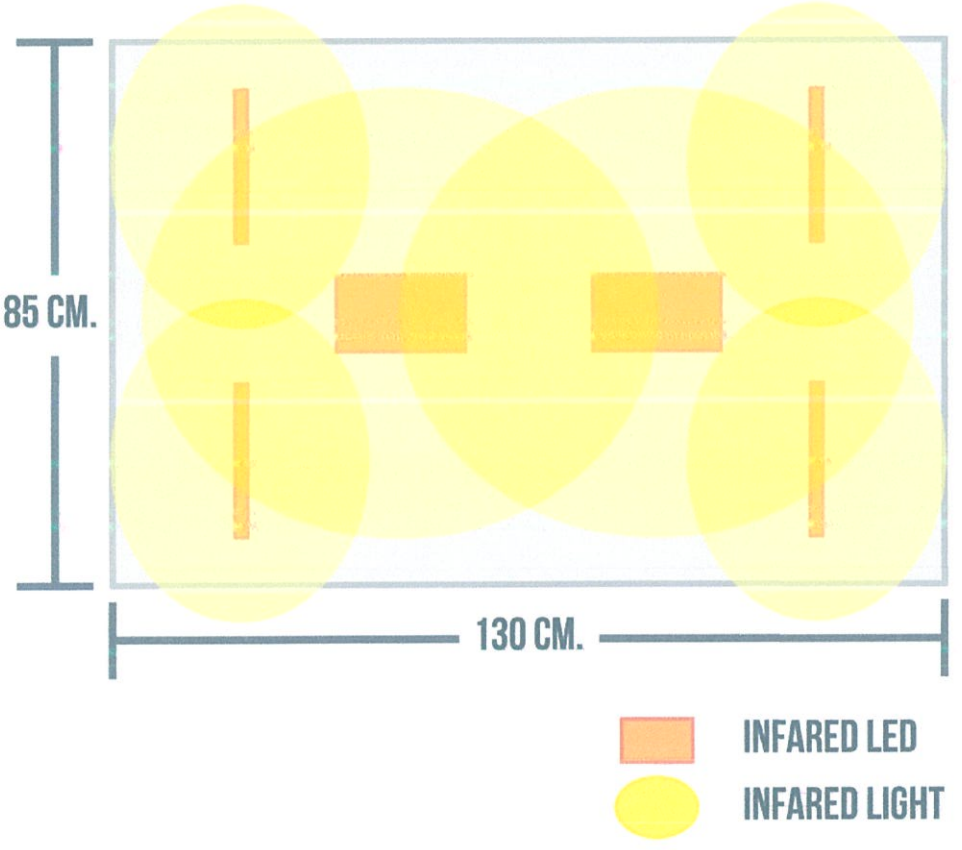


รูปที่ 3.7 โครงสร้างของโต๊ะอัจฉริยะ

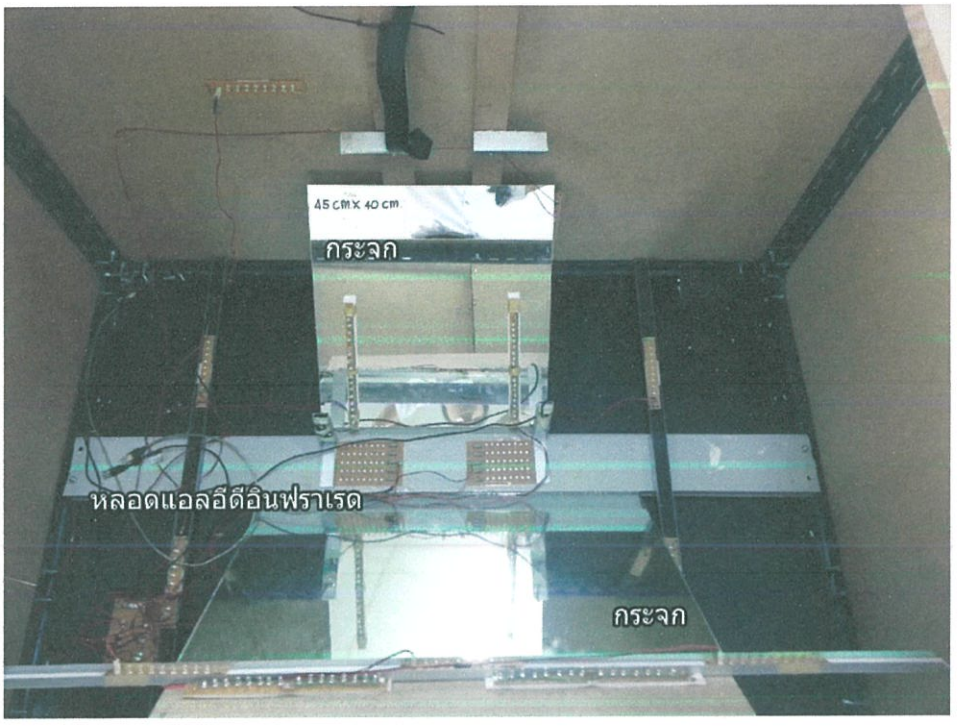
3.2.2 หน้าจอสัมผัสเทคโนโลยีอินฟราเรด เทคนิค Rear Diffused Illumination (Rear DI)

โต๊ะอัจฉริยะใช้เทคโนโลยีหน้าจอสัมผัสอินฟราเรด เทคนิค Rear Diffused Illumination(Rear DI) เนื่องจากเป็นเทคโนโลยีที่เหมาะสำหรับการทำโต๊ะขนาดใหญ่สามารถหาอุปกรณ์ในการทำได้ง่ายและประสิทธิภาพสามารถยอมรับได้ โดยเทคโนโลยีหน้าจอสัมผัสอินฟราเรด เทคนิค Rear DI นี้ใช้หลักการให้แสงอินฟราเรดด้านล่างของหน้าจอสัมผัสมีมากกว่าด้านบนของหน้าจอสัมผัส ทำให้เมื่อมีการสัมผัสจะเกิดการหักเหและเกิดความต่างของแสงทำให้ภาพที่ได้จากกล้องอินฟราเรดสามารถนำไประบุตำแหน่งของการสัมผัสได้

จากที่กล่าวข้างต้นเทคนิค Rear DI ต้องทำให้แสงอินฟราเรดด้านล่างของหน้าจอสัมผัสมีมากกว่าด้านบนของหน้าจอสัมผัสการออกแบบโต๊ะจึงต้องเป็นโต๊ะปิดเพื่อช่วยให้แสงกระจายอยู่ทั่วด้านล่างของหน้าจอสัมผัส และการวางแอลอีดีอินฟราเรดเนื่องจากหลอดแอลอีดีที่มีอยู่จำกัดจึงออกแบบโดยการวางแอลอีดีเป็นแผงยาวรอบๆและฉายขึ้นไปยังหน้าจอเพื่อให้แสงไปถึงหน้าจอได้ง่ายและใช้กระจกสะท้อนแสงของหลอดLEDเพื่อช่วยในการกระจายแสง มีการวางหลอดแอลอีดีเพื่อฉายขึ้นไปยังหน้าจอตรงกลางที่แอลอีดีที่วางรอบๆนั้นส่องไม่ถึง ดังรูปที่ 3.8 และรูปที่ 3.9



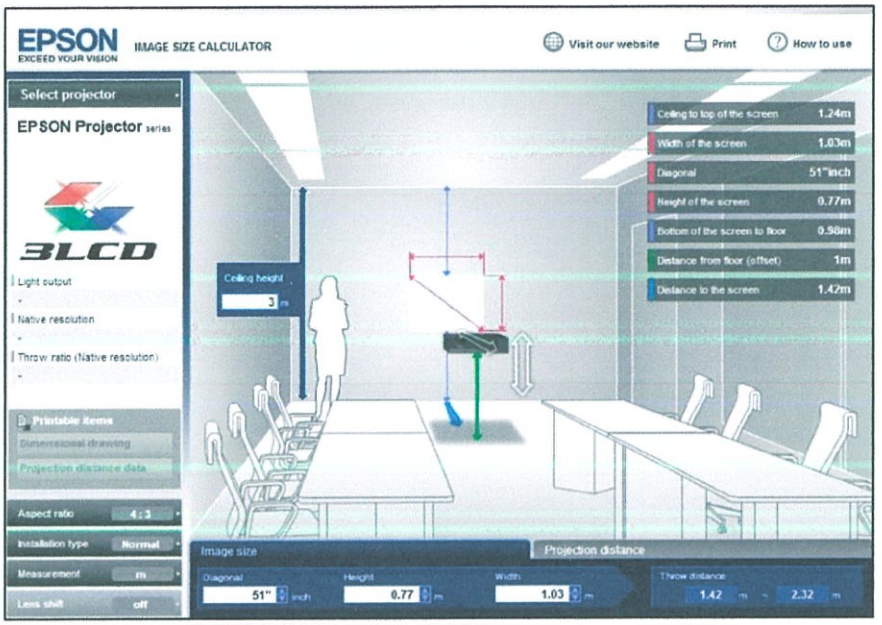
รูปที่ 3.8 การวางแอลอีดีอินฟราเรด (ในมุมมองด้านบน (Top view))



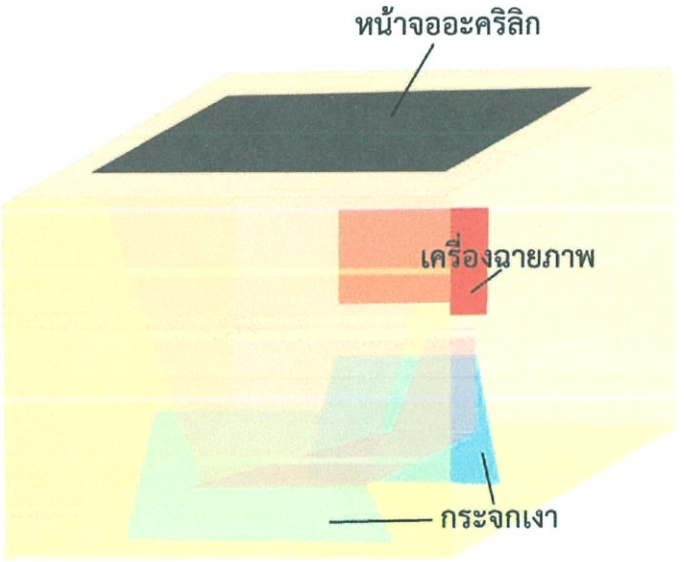
รูปที่ 3.9 การวางแอลอีดีอินฟราเรด (ในมุมมองด้านบน (Top view))

3.2.3 การสะท้อนเครื่องฉายภาพ

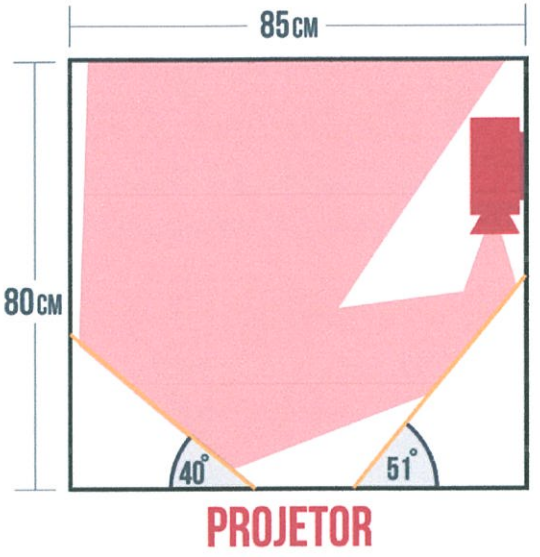
เนื่องจากเครื่องฉายภาพมีระยะการฉายภาพของแต่ละรุ่นของเครื่องฉายภาพ โดยเครื่องฉายภาพที่ใช้ EPSONBusinessEB-S02 สามารถหาระยะฉายภาพโดยการคำนวณจากhttp://www.epson.com/ ดังรูปที่ 3.9 ซึ่งค่าที่ได้เป็นค่าระยะทางประมาณ 1.4 เมตรถึง 2.3 เมตร ซึ่งความสูงของโต๊ะที่ออกแบบนั้นสูง 80 เซนติเมตรจึงต้องทำการสะท้อนภาพจากเครื่องฉายภาพโดยใช้กระจกเงาเพื่อเป็นการเพิ่มระยะให้สามารถฉายภาพได้ ซึ่งสามารถออกแบบได้ดังรูปที่ 3.10 3.11 และ 3.12



รูปที่ 3.10 ระยะฉายภาพโดยการคำนวณจาก
http://www.epson.com/alf_upload/landing/distance-calculator/



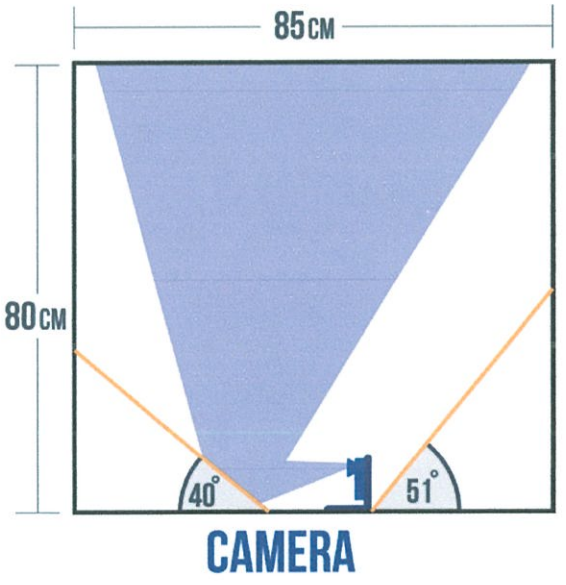
รูปที่ 3.11 การสะท้อนภาพจากเครื่องฉายภาพโดยใช้กระจกเงาเพื่อเป็นการเพิ่มระยะให้สามารถฉายภาพได้



รูปที่ 3.12 มุมการสะท้อนเครื่องฉายภาพโดยใช้กระจกเงาเมื่อมองภาพตัดขวาง (ในมุมมองด้านข้าง (Side view))

3.2.4 การสะท้อนกล้องเว็บแคมอินฟราเรด

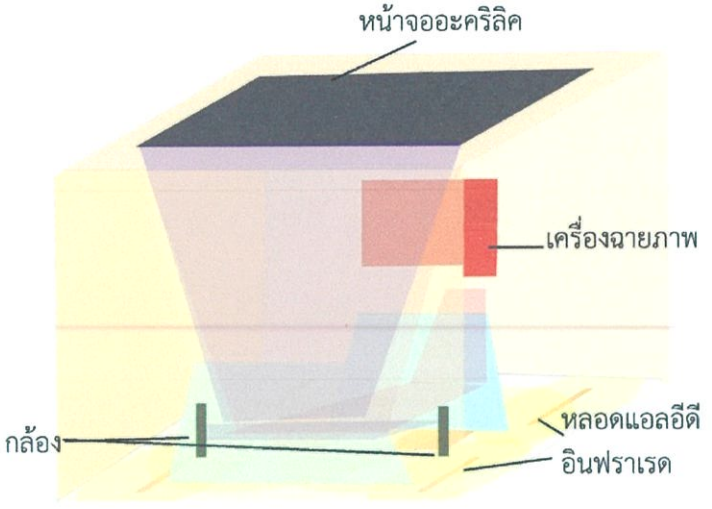
เนื่องจากกล้อง Logitech C170 มีความละเอียดภาพที่ค่อนข้างต่ำการรับภาพจากหน้าจอที่มีขนาดใหญ่ด้วยกล้องเพียงตัวเดียวไม่สามารถนำมาใช้กับเทคโนโลยีหน้าจอสัมผัสอินฟราเรดได้ จึงต้องทำการเพิ่มกล้องขึ้นมาอีกหนึ่งด้วย โดยแต่ละตัวทำหน้าที่รับภาพจากครึ่งหนึ่งของหน้าจอ และการจับภาพยังคงจับได้ไม่เต็มหน้าจอ จึงต้องมีการเพิ่มระยะการรับภาพโดยการรับภาพจากกระจกที่เกิดการสะท้อนกลับมา ดังรูปที่ 3.13



รูปที่ 3.13 การสะท้อนการรับภาพของกล้องเว็บแคมอินฟราเรดเมื่อมองภาพตัดขวาง (ในมุมมองด้านข้าง (Side view))

3.2.5 โครงสร้างโดยรวมของฮาร์ดแวร์

โครงสร้างโดยรวมของโต๊ะอัจฉริยะสามารถแสดงได้ดังรูปที่ 3.14 รูปที่ 3.15 และรูปที่ 3.16



รูปที่ 3.14 แบบโครงสร้างภายในของโต๊ะอัจฉริยะ

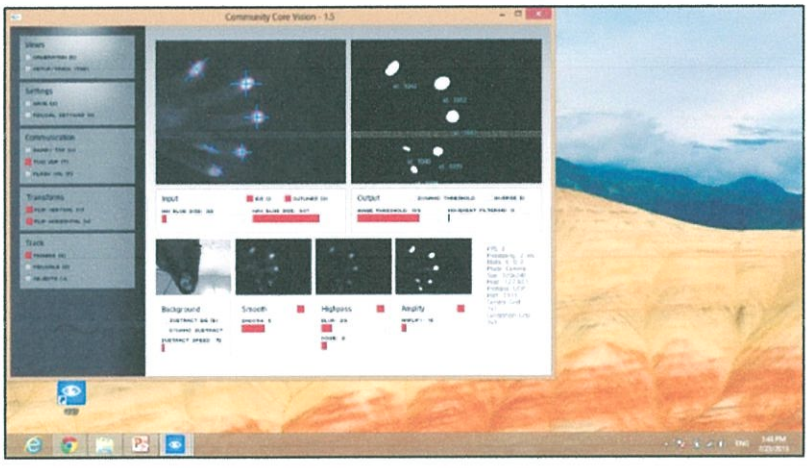
รูปที่ 3.15 โครงสร้างภายในของโต๊ะอัจฉริยะ ในมุมมองด้านบน และ ด้านข้าง

รูปที่ 3.16 แบบโดยรวมของโต๊ะอัจฉริยะ

3.3 การออกแบบซอฟต์แวร์

3.3.1 การตั้งค่าเพื่อการระบุตำแหน่งการสัมผัสหน้าจอ

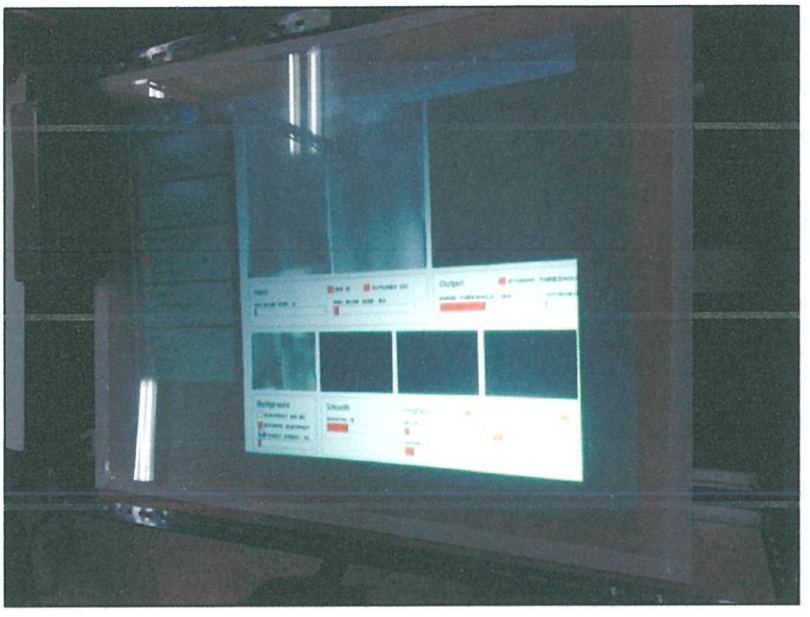
โปรแกรม Community Core Vision (CCV) ใช้รับอินพุตจากวีดีโอจากกล้องเว็บแคมอินฟราเรด ทำการปรับค่า Amplify, Smooth, Highpass และ Threshold เพื่อให้ได้ความแม่นยำในการสัมผัสดังรูปที่ 3.17



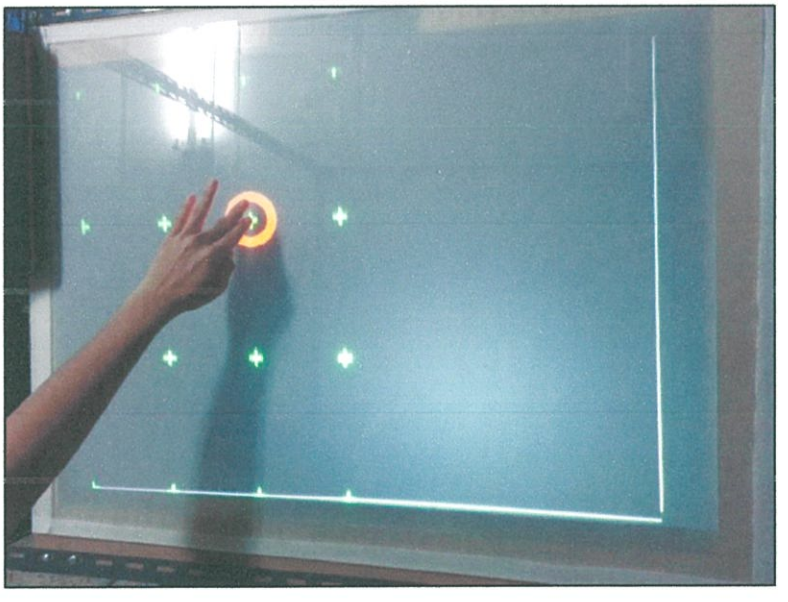
รูปที่ 3.17 การปรับค่าในโปรแกรมccv เพื่อให้ได้ความแม่นยำในการสัมผัส

เมื่อทำการปรับค่าที่ได้ความแม่นยำเมื่อมีการสัมผัสแล้วจะทำการปรับเทียบ (Calibration) เพื่อให้การสัมผัสนั้นตรงกับหน้าจอที่ใช้งานจริงดังรูปที่ 3.18 โดยการการปรับเทียบทำได้โดยสัมผัสตรงส่วนที่

โปรแกรมขึ้นจุดมาเมื่อรับข้อมูลตำแหน่งโดยเริ่มจากจุดที่มีวงกลมสีแดงและสัมผัสจนกว่าจะเปลี่ยนเป็นวงกลมสีฟ้าทั้งวงดังรูปที่ 3.19



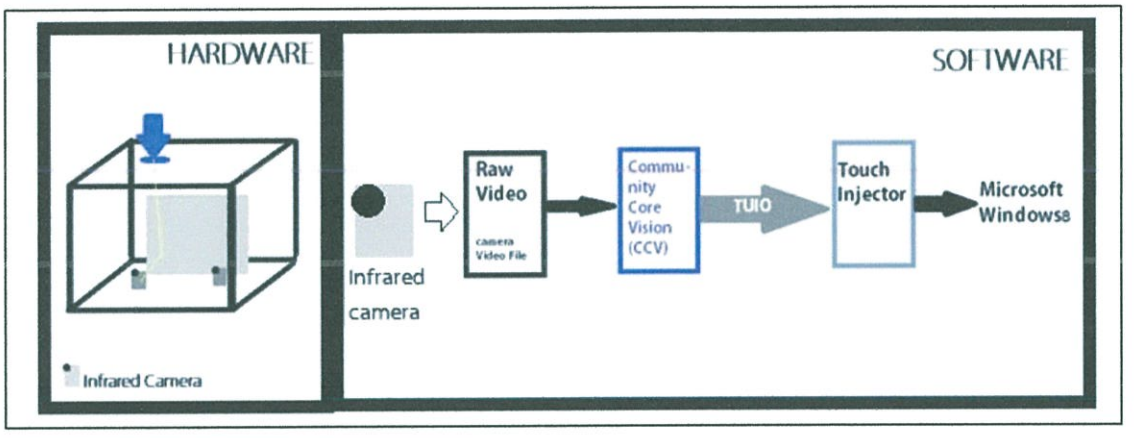
รูปที่ 3.18 การแสดงการปรับค่าที่ได้ความแม่นยำเมื่อมีการสัมผัส



รูปที่ 3.19 การทำการปรับเทียบเพื่อให้การสัมผัสตรงกับหน้าจอที่ใช้งานจริง

3.3.2 ซอฟต์แวร์แอพพลิเคชั่นที่สามารถใช้งานร่วมกับ Microsoft Windows 8

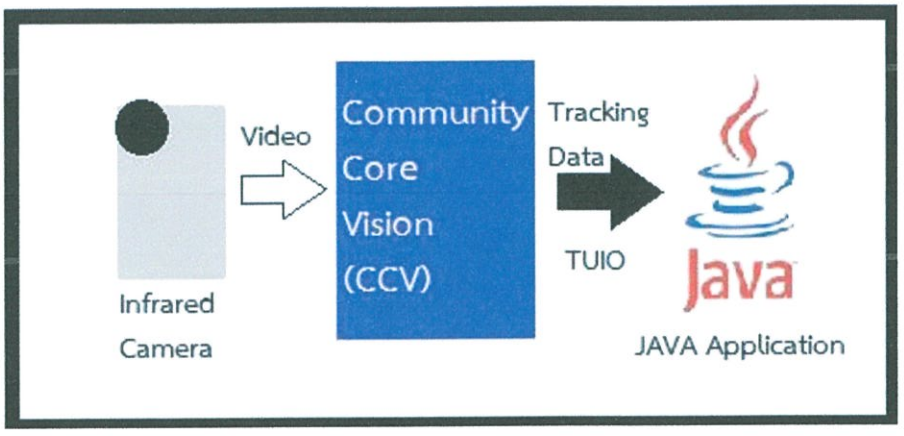
เมื่อทำการตั้งค่าและคาลริเบรชั่นเรียบร้อยทำการเลือกให้โปรแกรม Community Core Vision (CCV) ส่งข้อมูลผ่านโปรโตคอล TUIO เพื่อส่งข้อมูลการระบุตำแหน่งไปยังแอพพลิเคชั่น ซึ่งคือโปรแกรม Touch Injector ที่ทำหน้าสร้างการสัมผัสหน้าจอที่ได้รับมาจากโปรโตคอล TUIO สำหรับ Microsoft Windows 8 ทำให้สามารถเป็นคอมพิวเตอร์หน้าจอสัมผัสได้ และเครื่องฉายภาพทำหน้าที่เป็นการแสดงผลของผู้ใช้ อธิบายเป็นบล็อคไดอะแกรม ดังรูปที่ 3.20



รูปที่ 3.20 บล็อคไดอะแกรมการทำงานของซอฟต์แวร์ ที่สามารถใช้งานร่วมกับ Microsoft Windows8

3.3.3 ซอฟต์แวร์แอพพลิเคชั่นที่ทำงานร่วมกับภาษาจาวา

การทำงานร่วมกับแอพพลิเคชั่นที่เขียนด้วยภาษาจาวา ทำได้โดยการเขียนแอพพลิเคชั่นรับข้อมูลระบุตำแหน่งโดยใช้โปรโตคอล TUIO โดยใช้โปรแกรม CCV ทำการส่งผ่านข้อมูลการระบุตำแหน่งผ่านโปรโตคอล TUIO โดยพอร์ต 333 และส่งข้อมูลต่อมายังแอพพลิเคชั่นที่เขียนขึ้นสามารถแสดงได้ดังรูปที่ 3.21 ซึ่งสามารถรองรับการสัมผัสหลายรูปแบบได้



รูปที่ 3.21 บล็อคไดอะแกรมการทำงานของซอฟต์แวร์ ที่สามารถใช้งานร่วมกับแอ๊พพลิเคชั่นภาษาจาวา

แอพพลิเคชั่นออกแบบกตกแต่งภายในบ้าน (Interior Design)

แอพพลิเคชั่นออกแบบกตกแต่งภายในบ้าน เป็นแอพพลิเคชั่นในการนำเสนอการจัดวางเฟอร์นิเจอร์ต่างๆ ภายในบ้านเพื่อช่วยให้นักออกแบบและลูกค้าสามารถช่วยกันออกแบบการจัดวางเฟอร์นิเจอร์ต่างๆภายในบ้านได้

รูปที่ 3.22 แบบหน้าจออินเตอร์เฟสของแอพพลิเคชั่น

ส่วนของแอพพลิเคชั่นแบ่งเป็น 3 ส่วนคือส่วนของ คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น, คลาส Scene และ คลาสหลักในการเขียนแอพพลิเคชั่น

❖ คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น เพื่อใช้ในการเรียกใช้คลาสต่างๆในแพ็คเกจโดยเมื่อทำการสร้างคลาสจะตั้งส่วนนี้เป็นส่วนหลัก (Main) ดังรูป 3.23

StartInteriorDesign.java

```
package project.interiorDesign;

import org.mt4j.MTApplication;

public class StartInteriorDesign extends MTApplication {

    private static final long serialVersionUID = 1L;

    public static void main(String[] args) {
        initialize();
    }

    //@Override
    public void startUp() {
        this.addScene(new MainInteriorDesign(this, "Main interior design"));
    }
}
```

รูปที่ 3.23 คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น

❖ คลาส Scene เพื่อนำมารองรับพื้นหลังในการสร้างหน้าจอแอพพลิเคชั่น ดังรูปที่ 3.24

SceneInteriorDesign.java

```
package project.interiorDesign;

import org.mt4j.AbstractMTApplication;

public class SceneInteriorDesign extends AbstractScene {

    private AbstractMTApplication mtApp;
    private AbstractShape iconFurniture;
    private float stepDistance;
    private Vector3D localIconCenter;
    private float iconWidthHalf;
    private float iconHeightHalf;

    public SceneInteriorDesign(AbstractMTApplication mtApplication, String name) {
        super(mtApplication, name);
        this.mtApp = mtApplication;
    }

    public void setIcon(AbstractShape icon) {
        this.iconFurniture = icon;
        this.localIconCenter = iconFurniture.getCenterPointLocal();
        this.iconWidthHalf = iconFurniture.getWidthXY(TransformSpace.LOCAL)/2f;
        this.iconHeightHalf = iconFurniture.getHeightXY(TransformSpace.LOCAL)/2f;
        this.stepDistance = iconWidthHalf/2.0f;
    }

    public void onEnter() {

    }

    public void onLeave() {

    }
}
```

รูปที่ 3.24 คลาส Scene

❖ คลาสหลักในการเขียนแอพพลิเคชั่น

ส่วนแรกของคลาสหลักในการเขียนโปรแกรมนั้นคือการเรียกไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่น (โปรแกรมส่วนใหญ่ที่ใช้กับภาษาจาวาในปัจจุบันจะมีการนำเข้าไลบรารี่ให้อัตโนมัติหากมีการเรียกใช้คลาส ออบเจ็กต์และเมธอด ของไลบรารี่นั้น หากไลบรารี่นั้นอยู่ในโฟลเดอร์เดียวกันกับแพ็คเกจที่ได้เขียนโปรแกรม) ดังรูปที่ 3.25

MainInteriorDesign.java

```
package project.interiorDesign;

import org.mt4j.AbstractMTApplication;
import org.mt4j.components.TransformSpace;
import org.mt4j.components.visibleComponents.shapes.MTRectangle.PositionAnchor;
import org.mt4j.components.visibleComponents.shapes.MTRoundRectangle;
import org.mt4j.components.visibleComponents.widgets.MTBackgroundImage;
import org.mt4j.components.visibleComponents.widgets.MTList;
import org.mt4j.components.visibleComponents.widgets.MTListCell;
import org.mt4j.components.visibleComponents.widgets.MTSceneTexture;
import org.mt4j.components.visibleComponents.widgets.MTHandler;
import org.mt4j.components.visibleComponents.widgets.MTTextArea;
import org.mt4j.input.inputProcessors.IGestureEventListener;
import org.mt4j.input.inputProcessors.MTGestureEvent;
import org.mt4j.input.inputProcessors.componentProcessors.tapProcessor.TapEvent;
import org.mt4j.input.inputProcessors.componentProcessors.tapProcessor.TapProcessor;
import org.mt4j.sceneManagement.AbstractScene;
import org.mt4j.sceneManagement.IPreDrawAction;
import org.mt4j.util.MTColor;
import org.mt4j.util.font.FontManager;
import org.mt4j.util.font.IFont;
import org.mt4j.util.math.Vector3D;
```

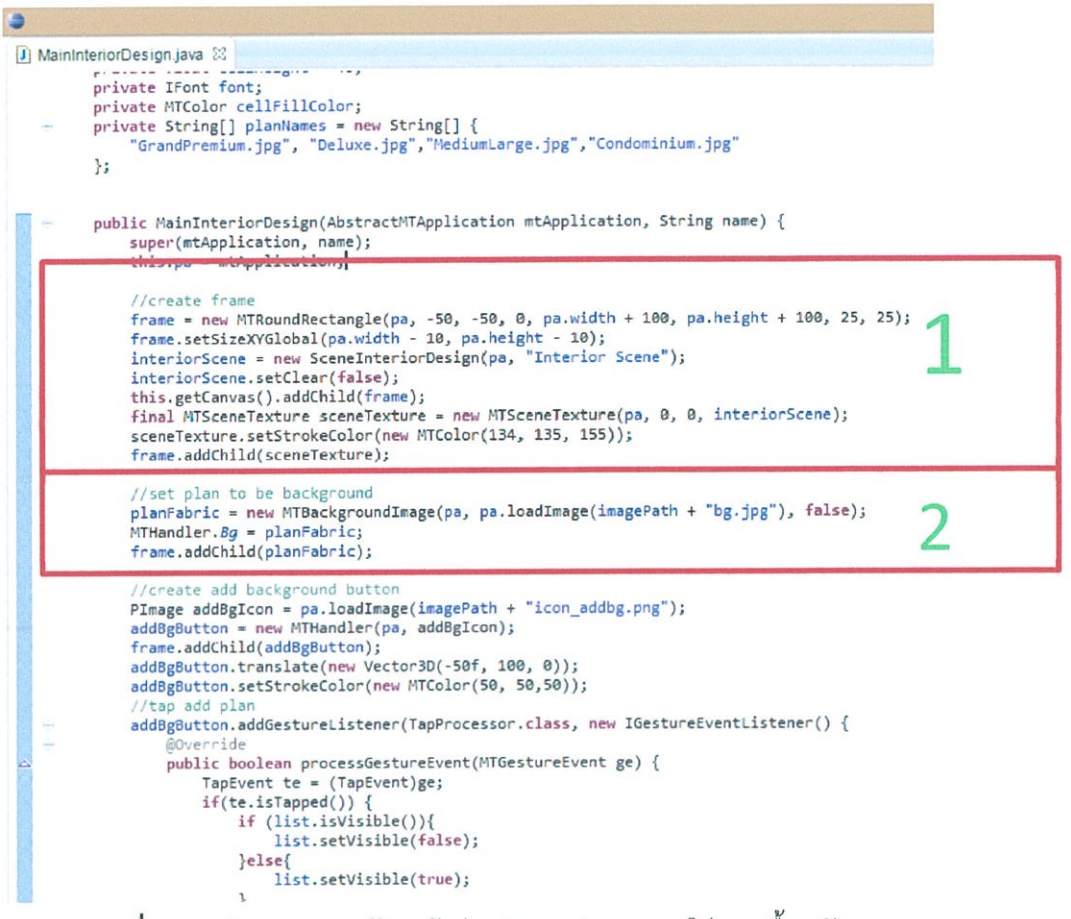
รูปที่ 3.25 ไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่น

การใช้คำสั่ง extends เพื่อนำคลาส AbstractScene ซึ่งเป็นคลาสในการสร้างหน้าจอเพื่อมารองรับพื้นหลังมาใช้ โดยคำสั่ง extends นี้จะนำคลาส AbstractScene มาทั้งคลาสเพื่อใช้สามารเรียกออบเจกต์และเมธอดในคลาสนั้นมาใช้ได้ ดังรูปที่ 3.26

```
public class MainInteriorDesign extends AbstractScene {

    private AbstractMTApplication pa;
    private String imagePath = "project" + AbstractMTApplication.separator + "interiorDesign" + AbstractMTApplication.separator + "data" + AbstractMTApplication.separator + "image" +
    private MTRoundRectangle frame;
    private MTBackgroundImage planFabric;
    private SceneInteriorDesign interiorScene;
    private MTList list;
    private MTHandler addBgButton;
    private float cellWidth = 150;
    private float cellHeight = 40;
    private IFont font;
    private MTColor cellFillColor;
    private String[] planNames = new String[] {
        "GrandPremium.jpg", "Deluxe.jpg","MediumLarge.jpg","Condominium.jpg"
    };
```

รูปที่ 3.26 การใช้คำสั่ง extends

รูปที่ 3.27 ส่วนที่ 1 คือของการสร้างหน้าต่างของแอพพลิเคชั่น การ addchild คือการให้แสดงผลบนหน้าแอพพลิเคชั่นที่สร้างขึ้น ส่วนที่ 2 เป็นส่วนของการสร้างภาพพื้นหลัง



รูปที่ 3.27 ส่วนของการสร้างหน้าต่าง (frame) และการใส่ภาพพื้นหลัง

เมื่อได้หน้าต่างของแอพลิเคชั่นต่อมาจะเป็นการสร้างปุ่มกดตามที่ได้ออกแบบไว้ โดยปุ่มกดแรกนั้นเป็นปุ่มกดเพื่อเลือกแบบ(Plan)ต่างๆ ซึ่งเมื่อกดปุ่มนี้จะต้องมีลิส(List) ของภาพแบบต่างออกมา ดังรูปที่ 3.28 และในลิสจะต้องมีการสร้างเซลล์ของลิส และเมื่อกดเซลล์ของลิสนั้นจะต้องทำการเปลี่ยนภาพพื้นหลังที่เป็นแบบบ้านต่างๆ ดังรูปที่ 3.29

```
//create add background button
PImage addBgIcon = pa.loadImage(imagePath + "icon_addbg.png");
addBgButton = new MTHandler(pa, addBgIcon);
frame.addChild(addBgButton);
addBgButton.translate(new Vector3D(-50f, 100, 0));
addBgButton.setStrokeColor(new MTColor(50, 50,50));
//tap add plan
addBgButton.addGestureListener(TapProcessor.class, new IGestureEventListener() {
    @Override
    public boolean processGestureEvent(MTGestureEvent ge) {
        TapEvent te = (TapEvent)ge;
        if(te.isTapped()) {
            if (list.isVisible()){
                list.setVisible(false);
            }else{
                list.setVisible(true);
            }
        }
        return false;
    }
});

//create List
font = FontManager.getInstance().createFont(pa, "SansSerif", 16, MTColor.BLACK, false);
cellFillColor = new MTColor(MTColor.WHITE);
list = new MTList(pa, addBgButton.getWidthXY(TransformSpace.GLOBAL) - 47f, 0, cellWidth+2, planNames.length * cellHeight + planNames.length*3);
list.setNoStroke(true);
list.setFillColor(MTColor.WHITE);
list.unregisterAllInputProcessors();
list.setAnchor(PositionAnchor.UPPER_LEFT);
list.setVisible(false);
for(String temp: planNames) {
    list.addListElement(this.createCell(pa, font, temp, cellFillColor, cellWidth, cellHeight));
};
frame.addChild(list);

//create Desk icon
PImage deskIcon = pa.loadImage(imagePath + "icon-officedesk.png");
final MTHandler deskButton = new MTHandler(pa, deskIcon);
frame.addChild(deskButton);
deskButton.translate(new Vector3D(-50f, 200, 0));
```

รูปที่ 3.28 ส่วนของการกดปุ่มจะแสดงลิสให้เห็น

```
private MTListCell createCell(final AbstractMTApplication mtApplication, IFont font, final String temp, final MTColor cellFillColor, float cellWidth, float cellHeight) {
    final MTListCell cell = new MTListCell(mtApplication, cellWidth, cellHeight);
    cell.setFillColor(cellFillColor);
    final MTTextArea listLabel = new MTTextArea(mtApplication, font);
    listLabel.setNoFill(true);
    listLabel.setNoStroke(true);
    listLabel.setText(temp);
    cell.addChild(listLabel);
    listLabel.setPositionRelativeToParent(cell.getCenterPointLocal());
    cell.unregisterAllInputProcessors();
    cell.registerInputProcessor(new TapProcessor(mtApplication, 15));
    cell.addGestureListener(TapProcessor.class, new IGestureEventListener() {
        @Override
        public boolean processGestureEvent(MTGestureEvent ge) {
            TapEvent te = (TapEvent)ge;
            switch (te.getTapID()) {
            case TapEvent.TAP_DOWN:
                cell.setFillColor(MTColor.FUCHSIA);
                break;
            case TapEvent.TAP_UP:
                cell.setFillColor(MTColor.GREEN);
                break;
            case TapEvent.TAPPED:
                cell.setFillColor(MTColor.YELLOW);
                list.setVisible(false);
                registerPreDrawAction(new IPreDrawAction() {
                    @Override
                    public void processAction() {
                        mtApplication.invokeLater(new Runnable() {
                            @Override
                            public void run() {
                                //MTHandler delete = new MTHandler(mtApplication, texture)
                                loadNewPlan(temp);
                            }
                        });
                    }
                    @Override
                    public boolean isLoop() {
                        return false;
                    }
                });
                break;
```

รูปที่ 3.29 ส่วนของการสร้างเซลล์ให้ลิส เมื่อมีการกดที่ลิสจะทำการเปลี่ยนภาพพื้นหลัง

```
        }
            return false;
        }
    });
    return cell;
}

private void loadNewPlan(String temp) {
    //remove plan and object
    frame.removeChild(planFabric);
    frame.removeChild(list);

    //add new plan
    planFabric = new MTBackgroundImage(pa, pa.loadImage(imagePath + temp), false);
    MTHandler.Bg = planFabric;
    frame.addChild(planFabric);

    //add new list
    list = new MTList(pa, addBgButton.getWidthXY(TransformSpace.GLOBAL) - 47f, 0, cellWidth+2, planNames.length * cellHeight + planNames.length*3);
    list.setNoStroke(true);
    list.setFillColor(MTColor.WHITE);
    list.unregisterAllInputProcessors();
    list.setAnchor(PositionAnchor.UPPER_LEFT);
    list.setVisible(false);
    for(String temp2: planNames) {
        list.addListElement(this.createCell(pa, font, temp2, cellFillColor, cellWidth, cellHeight));
    };
    frame.addChild(list);
}

public void onEnter() {

}

public void onLeave() {

}
```

รูปที่ 3.30 การสร้างเซลล์ให้ลิส เมื่อมีการกดที่ลิสจะทำการเปลี่ยนภาพพื้นหลัง(ต่อ)

ส่วนปุ่มกดอื่นๆนั้นจะเป็นในรูปแบบเดียวกันคือเมื่อกดแล้วจะมีรูปไฟล์.svg ออกมาดังรูปที่ 3.31 และส่วนที่ทำให้รูปสามารถแสดงได้คือส่วนของคลาส MTHandler ซึ่งส่วนนี้จะทำหน้าที่ให้รูปแสดงและเก็บค่าของตัวมันเองไว้เพื่อการลบ โดยการลบจะทำโดยการกดค้างไว้เมื่อค้างไว้จนครบเวลาที่กำหนดก็จะลบตัวมันเองออก ดังรูปที่ 3.32

```
//create Desk icon
PImage deskIcon = pa.loadImage(imagePath + "icon-officedesk.png");
final MTHandler deskButton = new MTHandler(pa, deskIcon);
frame.addChild(deskButton);
deskButton.translate(new Vector3D(-50f, 200, 0));
deskButton.setStrokeColor(new MTColor(50, 50, 50));
//tap Desk
deskButton.setSvgPath("desksvg.svg");
deskButton.addGestureListener(TapProcessor.class, new IGestureEventListener() {
    @Override
    public boolean processGestureEvent(MTGestureEvent ge) {
        TapEvent te = (TapEvent)ge;
        if(te.isTapped()) {
            deskButton.addSvgImage(pa);
        }
        return false;
    }
});

//create sofa icon
PImage sofaIcon = pa.loadImage(imagePath + "icon-sofa.png");
final MTHandler sofaButton = new MTHandler(pa, sofaIcon);
frame.addChild(sofaButton);
sofaButton.translate(new Vector3D(-50f, 300, 0));
sofaButton.setStrokeColor(new MTColor(50, 50, 50));
//Tap sofa icon
sofaButton.setSvgPath("sofasvg.svg");
sofaButton.addGestureListener(TapProcessor.class, new IGestureEventListener() {
    @Override
    public boolean processGestureEvent(MTGestureEvent ge) {
        TapEvent te = (TapEvent)ge;
        if(te.isTapped()) {
            sofaButton.addSvgImage(pa);
        }
        return false;
    }
});
```

รูปที่ 3.31 การสร้างปุ่มกดเมื่อกดแล้วรูปไฟล์ .svg จะแสดง

```
+import org.mt4j.AbstractMTApplication;

public class MTHandler extends MTImageButton {

    private String imagePath = "project" + AbstractMTApplication.separator + "interiorDesign" + AbstractMTApplication.separator + "data" + AbstractMTApplication.separator + "image" +
    public static MTBackgroundImage Bg;
    public MTHandler(PApplet pApplet, PImage texture) {
        super(pApplet, texture);
    }

    public void addSvgImage(AbstractMTApplication app) {
        final MTSvg svgImage = new MTSvg(app, imagePath + getSvgPath());
        svgImage.setPositionGlobal(new Vector3D(app.width/5, app.height/5, 0));
        Bg.addChild(svgImage);
        svgImage.registerInputProcessor(new TapAndHoldProcessor(app, 2000));
        svgImage.addGestureListener(TapAndHoldProcessor.class, new TapAndHoldVisualizer(app, getParent()));
        svgImage.addGestureListener(TapAndHoldProcessor.class, new IGestureEventListener() {
            @Override
            public boolean processGestureEvent(MTGestureEvent geh) {
                TapAndHoldEvent tah = (TapAndHoldEvent)geh;
                switch(tah.getId()) {
                case TapAndHoldEvent.GESTURE_STARTED:
                    break;
                case TapAndHoldEvent.GESTURE_UPDATED:
                    break;
                case TapAndHoldEvent.GESTURE_ENDED:
                    if(tah.isHoldComplete()) {
                        svgImage.removeAllChildren();
                    }
                }
                return false;
            }
        });
```

รูปที่ 3.32 ส่วน MTHandler

แอพพลิเคชั่นวาดเขียน (Painting)

แอพพลิเคชั่นวาดเขียนเป็นแอพพลิเคชั่นที่สามารถวาดเขียนและลบได้ โดยการลบนั้นสามารถเลือกลบหรือเลือกล้างหน้าจอทั้งหมดได้

รูปที่ 3.33 แอพพลิเคชั่นวาดเขียน

ส่วนของแอพพลิเคชั่นแบ่งเป็น 3 ส่วนคือส่วนของ คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น, คลาส Scene และ คลาสหลักในการเขียนแอพพลิเคชั่น

❖ คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น

```
package project.painting;

import org.mt4j.MTApplication;

public class Startpainting extends MTApplication {
    private static final long serialVersionUID = 1L;

    public static void main(String[] args) {
        initialize();
        System.out.println("success!"); //DEBUG
    }

    public void startUp() {
        this.addScene(new Mainpainting(this,"Main painting"));


    }

}
```

รูปที่ 3.34 คลาสเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น

❖ คลาส Scene

ในส่วนของแอพพลิเคชั่นวาดเขียนนี้ ส่วนของคลาส Scene จะทำหน้าที่สร้างหน้าจอแอพพลิเคชั่น และทำหน้าที่รับตำแหน่งการสัมผัสแล้วทำการสร้างการเขียนที่คลาสนี้ดังรูปที่ 3.35,3.36 และ 3.37 โดยมีเมธอด setBrush คลาสหลักจะเรียกใช้เพื่อให้เกิดการแสดงรูป (เมื่อทำการแสดงรูปต่อๆกันไปเรื่อยๆจะเกิดเป็นเส้นเช่นเดียวกับการสัมผัสแล้วลากจะเกิดการสร้างรูปเรื่อยๆจนกลายเป็นเส้น) เมื่อสัมผัสหน้าจอส่วนต่างๆ ดังรูปที่ 3.38

```
package project.painting;

import java.util.HashMap;

import org.mt4j.AbstractMTApplication;
import org.mt4j.components.TransformSpace;
import org.mt4j.components.visibleComponents.shapes.AbstractShape;
import org.mt4j.input.IMTInputEventListener;
import org.mt4j.input.inputData.AbstractCursorInputEvt;
import org.mt4j.input.inputData.InputCursor;
import org.mt4j.input.inputData.MTInputEvent;
import org.mt4j.sceneManagement.AbstractScene;
import org.mt4j.sceneManagement.IPreDrawAction;
import org.mt4j.util.MTColor;
import org.mt4j.util.math.ToolsMath;
import org.mt4j.util.math.Vector3D;

import processing.core.PApplet;

public class paintingScene extends AbstractScene {
    private AbstractMTApplication mtApp;

    private AbstractShape drawShape;

    private float stepDistance;

    private Vector3D localBrushCenter;

    private float brushWidthHalf;

    private HashMap<InputCursor, Vector3D> cursorToLastDrawnPoint;

    private float brushHeightHalf;

    private float brushScale;

    private MTColor brushColor;

    private boolean dynamicBrush;
```

รูปที่ 3.35 ส่วนของคลาส Scene

```
    public paintingScene(AbstractMTApplication mtApplication, String name) {
        super(mtApplication, name);
        this.mtApp = mtApplication;

        this.getCanvas().setDepthBufferDisabled(true);

        /*
        this.drawShape = getDefaultBrush();
        this.localBrushCenter = drawShape.getCenterPointLocal();
        this.brushWidthHalf = drawShape.getWidthXY(TransformSpace.LOCAL)/2f;
        this.brushHeightHalf = drawShape.getHeightXY(TransformSpace.LOCAL)/2f;
        this.stepDistance = brushWidthHalf/2.5f;
        */

        this.brushColor = new MTColor(0,0,0);
        this.brushScale = 1.0f;
        this.dynamicBrush = true;
//      this.stepDistance = 5.5f;

    this.cursorToLastDrawnPoint = new HashMap<InputCursor, Vector3D>();

        this.getCanvas().addInputListener(new IMTInputEventListener() {
            public boolean processInputEvent(MTInputEvent inEvt){
                if(inEvt instanceof AbstractCursorInputEvt){
                    final AbstractCursorInputEvt posEvt = (AbstractCursorInputEvt)inEvt;
                    final InputCursor m = posEvt.getCursor();
//                  System.out.println("PrevPos: " + prevPos);
//                  System.out.println("Pos: " + pos);

                    if (posEvt.getId() != AbstractCursorInputEvt.INPUT_ENDED){
                        registerPreDrawAction(new IPreDrawAction() {
                            public void processAction() {
                                boolean firstPoint = false;
                                Vector3D lastDrawnPoint = cursorToLastDrawnPoint.get(m);
                                Vector3D pos = new Vector3D(posEvt.getX(), posEvt.getY(), 0);

                                if (lastDrawnPoint == null){
                                    lastDrawnPoint = new Vector3D(pos);
                                    cursorToLastDrawnPoint.put(m, lastDrawnPoint);
                                    firstPoint = true;
                                }else{
                                    if (lastDrawnPoint.equalsVector(pos))

                                        return;
                                }

                                float scaledStepDistance = stepDistance*brushScale;

                                Vector3D direction = pos.getSubtracted(lastDrawnPoint);
                                float distance = direction.length();
                                direction.normalizeLocal();
                                direction.scaleLocal(scaledStepDistance);

                                float howManySteps = distance/scaledStepDistance;
                                int stepsToTake = Math.round(howManySteps);

                                //Force draw at 1st point
                                if (firstPoint && stepsToTake == 0){
                                    stepsToTake = 1;
                                }
//                              System.out.println("Steps: " + stepsToTake);

//                              GL gl = Tools3D.getGL(mtApp);
//                              gl.glBlendFuncSeparate(GL.GL_SRC_ALPHA, GL.GL_ONE_MINUS_SRC_ALPHA, GL.GL_ONE, GL.GL_ONE_MINUS_SRC_ALPHA);

                                mtApp.pushMatrix();
                                //We would have to set up a default view here for stability? (default cam etc?)
                                getSceneCam().update();

                                Vector3D currentPos = new Vector3D(lastDrawnPoint);
                                for (int i = 0; i < stepsToTake; i++) { //start i at 1? no, we add first step at 0 already
                                    currentPos.addLocal(direction);
                                    //Draw new brush into FBO at correct position
                                    Vector3D diff = currentPos.getSubtracted(localBrushCenter);

                                    mtApp.pushMatrix();
                                    mtApp.translate(diff.x, diff.y);

                                    //NOTE: works only if brush upper left at 0,0
                                    mtApp.translate(brushWidthHalf, brushHeightHalf);
                                    mtApp.scale(brushScale);

                                    if (dynamicBrush){
                                    //Rotate brush randomly
//                                  mtApp.rotateZ(PApplet.radians(Tools3D.getRandom(0, 179)));
```

รูปที่ 3.36 ส่วนของคลาส Scene (ต่อ)

```
                        mtApp.rotateZ(PApplet.radians(ToolsMath.getRandom(-25, 25)));
//                      mtApp.rotateZ(PApplet.radians(Tools3D.getRandom(-9, 9)));
                        mtApp.translate(-brushWidthHalf, -brushHeightHalf);
                        }

                        /*
                        //Use random brush from brushes
                        int brushIndex = Math.round(Tools3D.getRandom(0, brushes.length-1));
                        AbstractShape brushToDraw = brushes[brushIndex];
                         */
                        AbstractShape brushToDraw = drawShape;

                        //Draw brush
                        brushToDraw.drawComponent(mtApp.g);

                        mtApp.popMatrix();
                    }
                    mtApp.popMatrix();

                    cursorToLastDrawnPoint.put(m, currentPos);
                }

                public boolean isLoop() {
                    return false;
                }
            });
        }else{
            cursorToLastDrawnPoint.remove(m);
        }
    }
    return false;
  }
});

}
```

รูปที่ 3.37 ส่วนของคลาส Scene (ต่อ)

```
    public void setBrush(AbstractShape brush){
        this.drawShape = brush;
        this.localBrushCenter = drawShape.getCenterPointLocal();
        this.brushWidthHalf = drawShape.getWidthXY(TransformSpace.LOCAL)/2f;
        this.brushHeightHalf = drawShape.getHeightXY(TransformSpace.LOCAL)/2f;
        this.stepDistance = brushWidthHalf/2.8f;
        this.drawShape.setFillColor(this.brushColor);
        this.drawShape.setStrokeColor(this.brushColor);
    }

    public void setBrushColor(MTColor color){
        this.brushColor = color;
        if (this.drawShape != null){
            drawShape.setFillColor(color);
            drawShape.setStrokeColor(color);
        }
    }

    public void setBrushScale(float scale){
        this.brushScale = scale;
    }

    public void onEnter() {

    }

    public void onLeave() {

    }
}
```

รูปที่ 3.38 เมธอดของคลาส Scene ที่สร้างรูปเมื่อสัมผัส

❖ คลาสหลักในการเขียนแอพพลิเคชั่น

ส่วนแรกของคลาสหลักในการเขียนโปรแกรมนั้นคือการเรียกไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่นดังรูปที่ 3.39

Mainpainting.java

```
package project.painting;

import org.mt4j.AbstractMTApplication;
import org.mt4j.components.visibleComponents.shapes.MTRectangle;
import org.mt4j.components.visibleComponents.shapes.MTRoundRectangle;
import org.mt4j.components.visibleComponents.widgets.MTHandler;
import org.mt4j.components.visibleComponents.widgets.MTList;
import org.mt4j.components.visibleComponents.widgets.MTListCell;
import org.mt4j.components.visibleComponents.widgets.MTSceneTexture;
import org.mt4j.components.visibleComponents.widgets.MTTextArea;
import org.mt4j.input.inputProcessors.IGestureEventListener;
import org.mt4j.input.inputProcessors.MTGestureEvent;
import org.mt4j.input.inputProcessors.componentProcessors.tapProcessor.TapEvent;
import org.mt4j.input.inputProcessors.componentProcessors.tapProcessor.TapProcessor;
import org.mt4j.sceneManagement.AbstractScene;
import org.mt4j.sceneManagement.IPreDrawAction;
import org.mt4j.util.MTColor;
import org.mt4j.util.font.IFont;
import org.mt4j.util.math.Vector3D;
import processing.core.PImage;
```

รูปที่ 3.39 ไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่น

สร้างหน้าต่างสำหรับการวาดเขียนดังรูปที่ 3.40

Mainpainting.java

```
package project.painting;

import org.mt4j.AbstractMTApplication;

public class Mainpainting extends AbstractScene {
    private AbstractMTApplication pa;
    private MTRoundRectangle frame;
    private paintingScene scenepainting;
    private String imagePath = "project" + AbstractMTApplication.separator + "painting" + AbstractMTApplication.separator + "data" + AbstractMTApplication.separator;
    private MTList list;

    public Mainpainting(AbstractMTApplication mtApplication, String name) {
        super(mtApplication, name);
        this.pa = mtApplication;

        //create frame
        frame = new MTRoundRectangle(pa, -50, -50, 0, pa.width + 100, pa.height + 100, 25, 25);
        frame.setSizeXYGlobal(pa.width - 50, pa.height - 10);
        frame.setFillColor(new MTColor(MTColor.WHITE));
        scenepainting = new paintingScene(pa, "Painting Scene");
        scenepainting.setClear(false);
        this.getCanvas().addChild(frame);

        final MTSceneTexture sceneTexture = new MTSceneTexture(pa, 0, 0, scenepainting);
        sceneTexture.setStrokeColor(new MTColor(MTColor.BLACK));
        frame.addChild(sceneTexture);
        sceneTexture.getFbo().clear(true, 255, 255, 255, 0, true);
```

รูปที่ 3.40 การสร้างหน้าต่างสำหรับการวาดเขียน

ส่วนของการใช้งานมีอยู่ 3 ส่วนคือส่วนของการล้างหน้าจอทั้งหมด ส่วนของการลบ และส่วนของการเขียน

ส่วนของการล้างหน้าจอทั้งหมดคือเมื่อกด จะทำการเซตหน้าจอทั้งหมดให้เป็นสีขาวเหมือนเดิมดังรูปที่ 3.41

```
//create RemoveAll ICON
PImage RemoveAllIcon = pa.loadImage(imagePath + "clear.png");
final MTHandler RemoveAllButton = new MTHandler(pa, RemoveAllIcon);
frame.addChild(RemoveAllButton);
RemoveAllButton.translate(new Vector3D(700f, 10f));
RemoveAllButton.setNoStroke(true);
RemoveAllButton.addGestureListener(TapProcessor.class, new IGestureEventListener() {
    @Override
    public boolean processGestureEvent(MTGestureEvent ge) {
        TapEvent te = (TapEvent)ge;
        if(te.isTapped()) {
            pa.invokeLater(new Runnable() {
                public void run() {
                    sceneTexture.getFbo().clear(true, 255, 255, 255, 0, true);
                }
            });
        }
        return false;
    }
});
```

รูปที่ 3.41 การล้างหน้าจอทั้งหมด

ส่วนของการลบและการเขียนจะเขียนเหมือนกันต่างกันตรงที่การลบคือการเขียนด้วยสีขาว โดยเมื่อทำการกดรูปที่กำหนดจะทำการส่งรูปที่เป็นลักษณะของการเขียน (รูปที่ Scene นำมาแสดง) ไปยังเมธอด Texture เพื่อทำการสร้างลักษณะการเขียน และทำการลงสีรูปที่นำมาแสดง (รูปที่นำมาแสดงจึงต้องเป็นสีขาวเพื่อให้สามารถลงสีทับได้) ดังรูปที่ 3.42 และรูปที่ 3.43

*Mainpainting.java

```
//create Eraser
PImage eraserIcon = pa.loadImage(imagePath + "eraser-icon.png");
final MTHandler eraserButton = new MTHandler(pa, eraserIcon);
frame.addChild(eraserButton);
eraserButton.translate(new Vector3D(400f, 550f));
eraserButton.setNoStroke(true);
eraserButton.addGestureListener(TapProcessor.class, new IGestureEventListener() {

    @Override
    public boolean processGestureEvent(MTGestureEvent ge) {
        TapEvent te = (TapEvent)ge;
        if(te.isTapped()){
            Texture("earse_brush.png");
            scenepainting.setBrushColor(new MTColor(MTColor.WHITE));
        }
        return false;
    }
});

//create Blue pen
        PImage blueIcon = pa.loadImage(imagePath + "blue-pen.png");
        final MTHandler blueButton = new MTHandler(pa, blueIcon);
        frame.addChild(blueButton);
        blueButton.translate(new Vector3D(550f, 500f));
        blueButton.setNoStroke(true);
        blueButton.addGestureListener(TapProcessor.class, new IGestureEventListener() {

            @Override
            public boolean processGestureEvent(MTGestureEvent ge) {
                TapEvent te = (TapEvent)ge;
                if(te.isTapped()){
                    Texture("minidot_brush.png");
                    scenepainting.setBrushColor(new MTColor(MTColor.BLUE));
                }
                return false;
            }
        });
//create Green pen
```

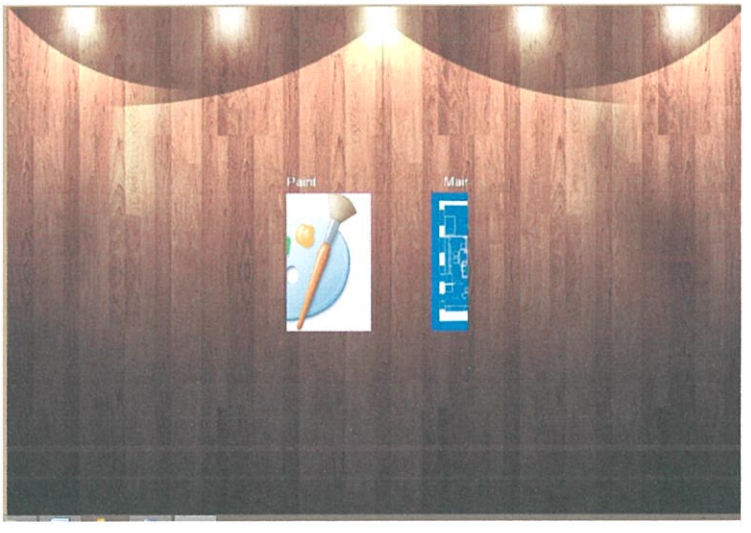
รูปที่ 3.42 ส่วนของการลบและการเขียน

```
// create texture
    private void  Texture(String temp){

          PImage textureImage = getMTApplication().loadImage(imagePath + temp);
          MTRectangle texture;
            texture = new MTRectangle(getMTApplication(),textureImage);
            texture.setPickable(false);
            texture.setNoFill(false);
            texture.setNoStroke(true);
            texture.setDrawSmooth(true);
            texture.setFillColor(new MTColor(0,0,0));
            scenepainting.setBrush(texture);

    }

}//end
```

รูปที่ 3.43 เมธอด Texture

หน้าเมนูหลัก

หน้าจอเมนูหลักทำหน้าที่รวบรวมแอพพลิเคชั่นต่างๆ ทำให้สามารถเข้าแอ๊พพลิเคชั่นต่างๆที่เขียนขึ้นได้จากหน้าจอนี้ดังรูปที่ 3.44



รูปที่ 3.44 แบบหน้าจอเมนูหลัก

แบ่งส่วนของแอพพลิเคชั่นเป็น 3 ส่วนคือส่วนของ คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น, คลาส Scene และ คลาสหลักในการเขียนแอพพลิเคชั่น

❖ คลาสในการเริ่มต้นเพื่อเรียกใช้แอพพลิเคชั่น

```
package project.menu;

import org.mt4j.MTApplication;

public class StartMenu extends MTApplication {
    private static final long serialVersionUID = 1L;

    public static void main(String[] args) {
        initialize();
        System.out.println("success!"); //DEBUG
    }

    @Override
    public void startUp() {
        this.addScene(new MenuScene(this, "Menu Scene"));
    }
}
```

รูปที่ 3.45 คลาสเริ่มต้นในการเรียกใช้แอพพลิเคชั่น

❖ คลาส Scene เพื่อนำมารองรับพื้นหลังในการสร้างหน้าจอแอพพลิเคชั่น ดังรูปที่ 3.46

```
package project.menu;

import org.mt4j.sceneManagement.Iscene;
public interface CreateScene {
    /**
     * Gets the new scene.
     *
     * @return the new scene
     */
    public Iscene getNewScene();
    /**
     * Gets the title.
     *
     * @return the title
     */
    public String getTitle();
}
```

รูปที่ 3.46 คลาส Scene

❖ คลาสหลักในการเขียนแอพพลิเคชั่น

ส่วนแรกของคลาสหลักในการเขียนโปรแกรมนั้นคือการเรียกไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่นดังรูปที่ 3.47

```
package project.menu;

import java.awt.event.KeyEvent;

import org.mt4j.AbstractMTApplication;
import org.mt4j.components.TransformSpace;
import org.mt4j.components.visibleComponents.shapes.MTPolygon;
import org.mt4j.components.visibleComponents.widgets.MTBackgroundImage;
import org.mt4j.components.visibleComponents.widgets.MTList;
import org.mt4j.components.visibleComponents.widgets.MTListCell;
import org.mt4j.components.visibleComponents.widgets.MTSceneMenu;
import org.mt4j.components.visibleComponents.widgets.MTSceneWindow;
import org.mt4j.components.visibleComponents.widgets.MTTextArea;
import org.mt4j.input.gestureAction.InertiaDragAction;
import org.mt4j.input.inputProcessors.IGestureEventListener;
import org.mt4j.input.inputProcessors.MTGestureEvent;
import org.mt4j.input.inputProcessors.componentProcessors.dragProcessor.DragProcessor;
import org.mt4j.input.inputProcessors.componentProcessors.tapProcessor.TapEvent;
import org.mt4j.input.inputProcessors.componentProcessors.tapProcessor.TapProcessor;
import org.mt4j.input.inputProcessors.globalProcessors.CursorTracer;
import org.mt4j.sceneManagement.AbstractScene;
import org.mt4j.sceneManagement.IScene;
import org.mt4j.sceneManagement.transition.BlendTransition;
import org.mt4j.util.MTColor;
import org.mt4j.util.font.FontManager;
import org.mt4j.util.font.IFont;
import org.mt4j.util.logging.ILogger;
import org.mt4j.util.logging.MTLoggerFactory;
import org.mt4j.util.math.Vector3D;
import org.mt4j.util.math.Vertex;
import org.mt4j.util.opengl.GLFBO;
```

รูปที่ 3.47 ไลบรารี่ที่ต้องใช้ในแอพพลิเคชั่น

ต่อมาเป็นการประกาศตัวแปรเพื่อสร้างหน้าต่างขึ้นมารองรับลิสของเมนูดังรูปที่ 3.48

```
    private static final ILogger logger = MTLoggerFactory.getLogger(MenuScene.class.getName());
    static {
        logger.setLevel(ILogger.INFO);
    }
    private AbstractMTApplication app;
    private boolean hasFBO;
    private MTList list;
    private IFont font;
    private int preferredIconHeight;
    private int gapBetweenIconAndReflection;
    private float displayHeightOfReflection;
    private float listWidth;
    private int listHeight;
    private int preferredIconWidth;
    private boolean switchDirectlyToScene = false;
    private MTBackgroundImage BG;
    private String imagePath = "project" + AbstractMTApplication.separator + "menu" + AbstractMTApplication.separator +

    public MenuScene(AbstractMTApplication mtApplication, String name) {
        super(mtApplication, name);

        this.app = mtApplication;
        this.hasFBO = GLFBO.isSupported(app);
        this.switchDirectlyToScene = !this.hasFBO || switchDirectlyToScene;
        this.registerGlobalInputProcessor(new CursorTracer(app, this));
        this.setClearColor(new MTColor(255, 102, 0, 255));  //BACKGROUND COLOUR
        BG = new MTBackgroundImage(app, app.loadImage(imagePath + "menu.jpg"), false);
        getCanvas().addChild(BG);
        preferredIconHeight = 192;
        preferredIconWidth = 256;
        gapBetweenIconAndReflection = 9;
        displayHeightOfReflection = preferredIconHeight*0.6f;
```

รูปที่ 3.48 การสร้างหน้าต่างของแอพพลิเคชั่น

สร้างลิสสำหรับนำไอคอนที่เรียกแอพพลิเคชั่นต่างๆไปใส่ดังรูปที่ 3.49

```
//CREATE LIST
listWidth = preferredIconHeight+displayHeightOfReflection+gapBetweenIconAndReflection;
//listHeight = app.width;
listHeight = 250;
list = new MTList(mtApplication,0, 0, listWidth, listHeight, 80);
list.setFillColor(new MTColor(150, 111, 120, 200)); //LIST COLOUR

list.setNoFill(true);
list.setNoStroke(true);
font = FontManager.getInstance().createFont(app, "SansSerif", 18, MTColor.WHITE);
```

รูปที่ 3.49 การสร้างลิส

นำไอคอนของแอพพลิเคชั่นต่างๆใส่ในลิสและเชื่อมโยงกับคลาสที่เรียกแอพพลิเคชั่นดังรูปที่ 3.50

```
    this.addScene(new CreateScene() {
        public Iscene getNewScene() {
            return new Mainpainting(app, "Main Painting");
        }
        public String getTitle() {
            return "Paint";
        }
    }, app.loadImage(this.getPathToIcons() + "icon-paint.png"));//Icon


    this.addScene(new CreateScene() {
        public Iscene getNewScene() {
            return new MainInteriorDesign(app, "Main interior design");
        }
        public String getTitle() {
            return "Main interior design";
        }
    }, app.loadImage(this.getPathToIcons() + "icon-in.png"));//Icon


    getCanvas().addChild(list);
    list.rotateZ(list.getCenterPointLocal(), -90, TransformSpace.LOCAL);
    list.setPositionGlobal(new Vector3D(app.width/2f, app.height/2f));
    getCanvas().setFrustumCulling(true);
}

private String getPathToIcons() {
    return "project"+AbstractMTApplication.separator+"menu"+AbstractMTApplication.separator+"data"+AbstractMTApplic
}
```

รูปที่ 3.50 การนำไอคอนแอพพลิเคชั่นใส่ในลิส

เมื่อทำการคลิกที่แอพพลิเคชั่นจะต้องสร้าง Scene ขึ้นมาเพื่อให้แสดงหน้าต่างของแอพพลิเคชั่นบนเมนู และหน้าต่างที่แสดงขึ้นมานั้นสามารถกดผิดและขยายเพื่อนเต็มหน้าจอได้ ดังรูปที่ 3.51 และ 3.52

```
private void addTapProcessor(MTListCell cell, final CreateScene createScene) {
    cell.registerInputProcessor(new TapProcessor(app, 15));
    cell.addGestureListener(TapProcessor.class, new IGestureEventListener() {
        public boolean processGestureEvent(MTGestureEvent ge) {
            TapEvent te = (TapEvent)ge;
            if (te.isTapped()) {
                final Iscene scene = createScene.getNewScene();
                if(!switchDirectlyToScene) {
                    if(hasFBO && scene instanceof AbstractScene) {
                        ((AbstractScene)scene).setTransition(new BlendTransition(app, 300));
                    }
                    final MTSceneWindow sceneWindow = new MTSceneWindow(app, scene, 100, 100);
                    sceneWindow.scaleGlobal(0.5f, 0.5f, 0.5f, sceneWindow.getCenterPointGlobal());
                    sceneWindow.addGestureListener(DragProcessor.class, new InertiaDragAction());
                    getCanvas().addChild(sceneWindow);
                }
                else {
                    float menuWidth = 64;
                    float menuHeight = 64;
                    MTSceneMenu sceneMenu = new MTSceneMenu(app, scene, app.width-menuWidth, app.height-menuHeight, menuWidth, menuHeight);
                    sceneMenu.addToScene();
                    app.addScene(scene);
                    app.pushScene();
                    app.changeScene(scene);
                }
            }
            return false;
        }
    });
}
```

รูปที่ 3.51 ส่วนของการคลิกจะสร้าง Scene ขึ้นมา

```
public void addScene(CreateScene sceneToCreate, PImage icon) {
    PImage reflection = this.getReflection(getMTApplication(), icon);
    float border = 1;
    float bothBorders = 2*border;
    float topShift = 30;
    float reflectionDistanceFromImage = topShift+gapBetweenIconAndReflection;
    float listCellWidth = listWidth;
    float realListCellWidth = listCellWidth-bothBorders;
    float listCellHeight = preferredIconHeight;
    MTListCell cell = new MTListCell(app, realListCellWidth, listCellHeight);
    cell.setNoFill(true);
    cell.setNoStroke(true);
    Vertex[] vertices = new Vertex[] {
            new Vertex(realListCellWidth-topShift, border, 0, 0, 0),
            new Vertex(realListCellWidth-topShift, listCellHeight-border, 0, 1, 0),
            new Vertex(realListCellWidth-topShift-icon.height, listCellHeight-border, 0, 1, 1),
            new Vertex(realListCellWidth-topShift-icon.height, border, 0, 0, 1),
            new Vertex(realListCellWidth-topShift, border, 0, 0, 0),
    };
    MTPolygon p = new MTPolygon(getMTApplication(), vertices);
    p.setTexture(icon);
    p.setStrokeColor(new MTColor(180, 80, 180, 255));
    Vertex[] verticesRef = new Vertex[] {
        new Vertex(listCellWidth-icon.height-reflectionDistanceFromImage, border, 0, 0, 0),
        new Vertex(listCellWidth-icon.height-reflectionDistanceFromImage, listCellHeight-border, 0, 1, 0),
        new Vertex(listCellWidth-icon.height-reflection.height-reflectionDistanceFromImage, listCellHeight-border, 0, 1, 1),
        new Vertex(listCellWidth-icon.height-reflection.height-reflectionDistanceFromImage, listCellHeight-border, 0, 0, 1),
        new Vertex(listCellWidth-icon.height-reflectionDistanceFromImage,                                    border, 0,  0,0),
    };
    MTPolygon pRef = new MTPolygon(getMTApplication(), verticesRef);
    pRef.setTexture(reflection);
    pRef.setNoStroke(true);
    cell.addChild(p);
    cell.addChild(pRef);
    list.addListElement(cell);
    addTapProcessor(cell, sceneToCreate);
    MTTextArea text = new MTTextArea(app, font);
    text.setFillColor(new MTColor(150, 150, 250, 200));
    text.setNoFill(true);
    text.setNoStroke(true);
    text.setText(sceneToCreate.getTitle());
    text.rotateZ(text.getCenterPointLocal(), 90, TransformSpace.LOCAL);
```

nuScene.java

```
    cell.addChild(text);
    text.setPositionRelativeToParent(cell.getCenterPointLocal());
    text.translate(new Vector3D(realListCellWidth*0.5f-text.getHeightXY(TransformSpace.LOCAL)*0.5f, 0));
}

private PImage getReflection(PApplet pa, PImage image) {
    int width = image.width*0; //no reflect
    int height = image.height*0; //no reflect
    PImage copyOfImage = pa.createImage(image.width, image.height, PApplet.ARGB);
    image.loadPixels();
    copyOfImage.loadPixels();
    for(int y = 0; y < height; y++) {
        for(int x = 0; x < width; x++) {
            int imageIndex = y*image.width+x;
            int currR = (image.pixels[imageIndex] >> 16) & 0xFF;
            int currG = (image.pixels[imageIndex] >> 8) & 0xFF;
            int currB = image.pixels[imageIndex] & 0xFF;
            int col = image.pixels[imageIndex];
            float alpha = pa.alpha(col);
            int reflectImageIndex = (image.height-y-1)*image.width+x;

            if(alpha <= 0.0f) {
                copyOfImage.pixels[reflectImageIndex] = pa.color(currR, currG, currB, 0.0f);
            }
            else {
                copyOfImage.pixels[reflectImageIndex] = pa.color(currR, currG, currB, Math.round(y*y*y*(0.00003f)-60));
            }
        }
    }
```

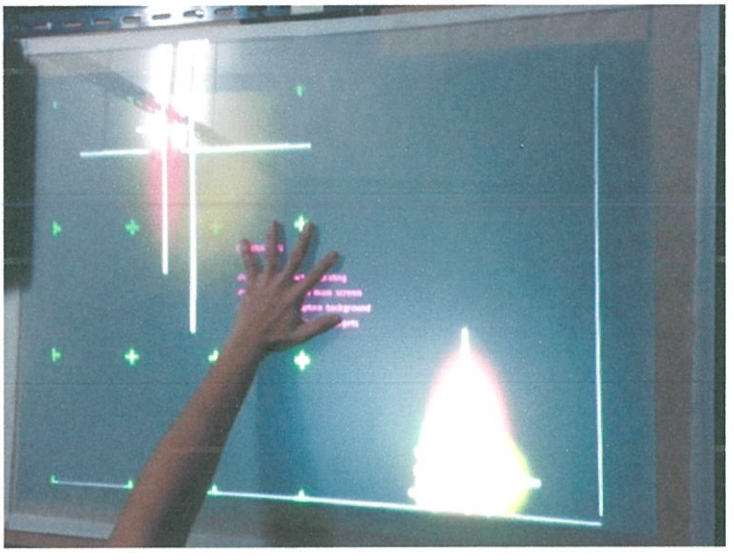
รูปที่ 3.52 รูปแบบหน้าต่าง Scene ที่แสดงขึ้นมา

บทที่ 4

ผลการทดลอง

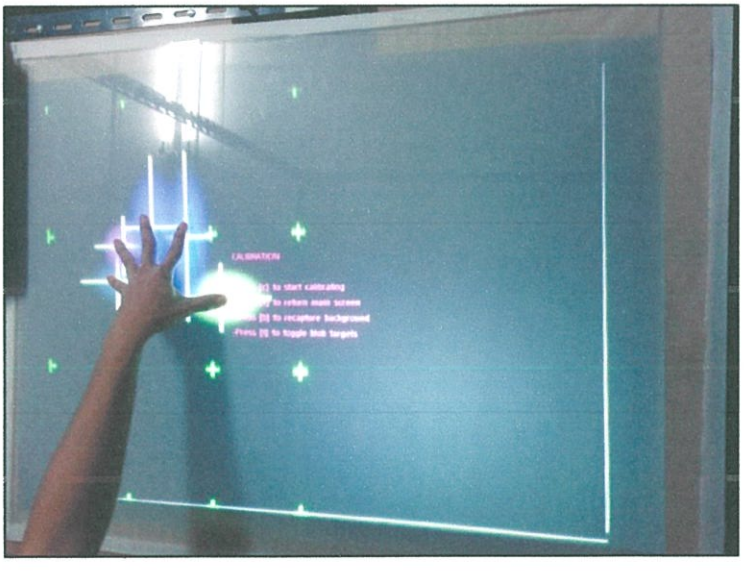
4.1 ผลการทำการปรับเทียบ (Calibration)

จากการทำการทดลองเมื่อได้ทำการตั้งค่าในโปรแกรม Community Core Vision (CCV) เรียบร้อย แต่ยังไม่ได้ทำการปรับเทียบ การสัมผัสบนหน้าจอสามารถสัมผัสได้แต่ไม่ตรงตามจุดสัมผัสที่ใช้ดังรูปที่ 4.1



รูปที่ 4.1 การสัมผัสหน้าจอที่ยังไม่ได้ทำการปรับเทียบ

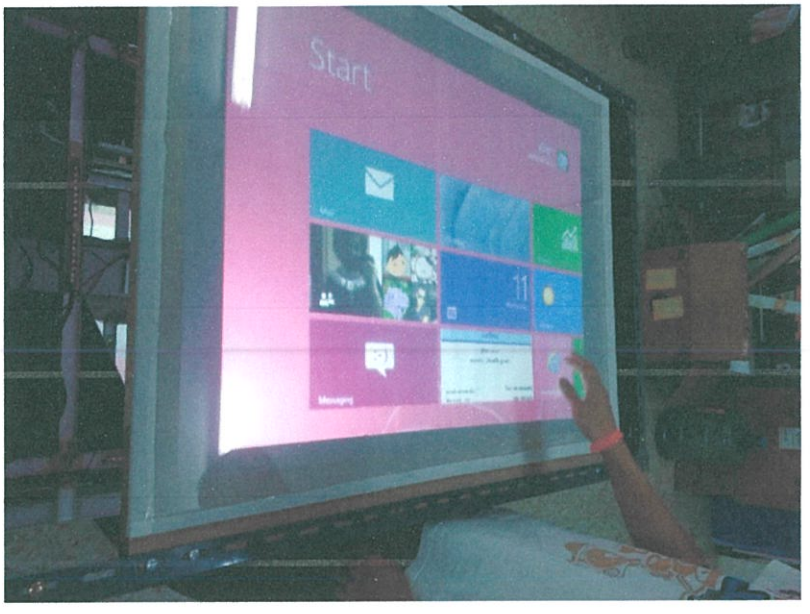
เมื่อทำการปรับเทียบจะทำให้การสัมผัสบนหน้าจอนั้นได้ตรงตามจุดที่ต้องการดังรูปที่ 4.2



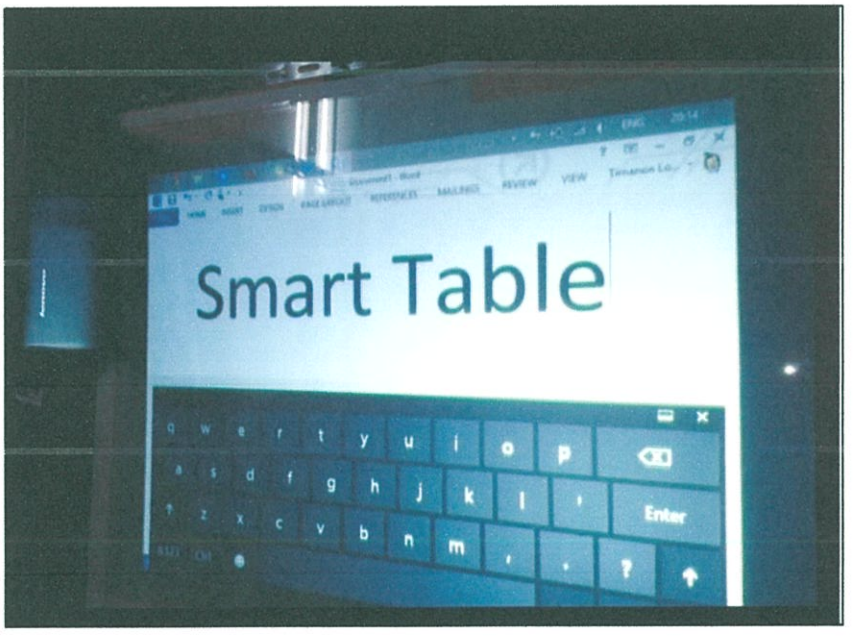
รูปที่ 4.2 การสัมผัสหน้าจอที่ทำการปรับเทียบเรียบร้อย

4.2 ผลการเชื่อมต่อโปรแกรม CCV กับ Touch Injector

เมื่อทำการตั้งค่าโปรแกรม CCV ให้ส่งข้อมูลการระบุตำแหน่งโดยโปรโตคอล TUIO ไปยังแอพพลิเคชั่น Touch Injector เพื่อให้โปรแกรม Touch Injector ทำการเชื่อมต่อกับ Microsoft Windows 8 ทำให้สามารถใช้เป็นคอมพิวเตอร์หน้าจอสัมผัสได้ ดังรูปที่ 4.3 – 4.8

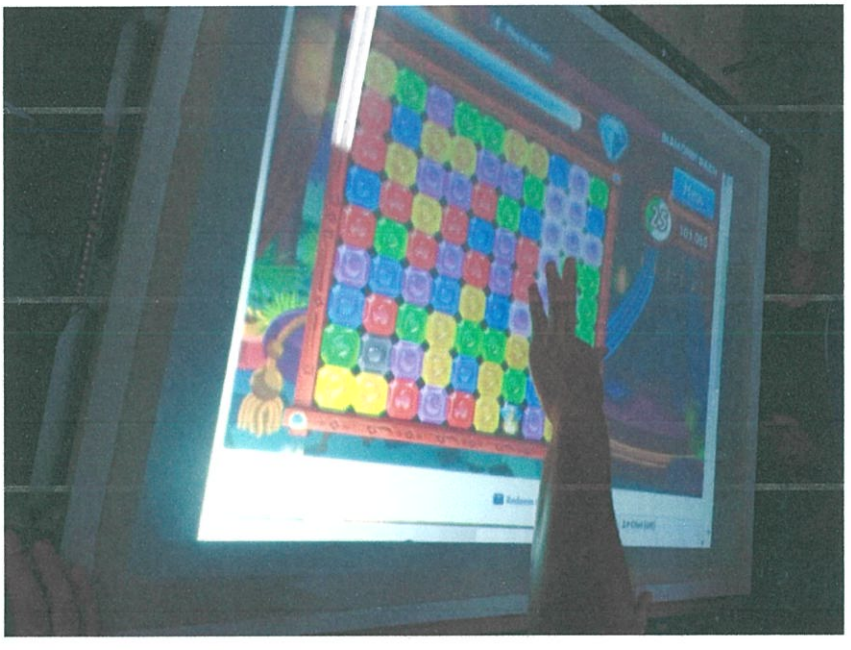


รูปที่ 4.3 การใช้โต๊ะอัจฉริยะร่วมกับ Microsoft Windows8

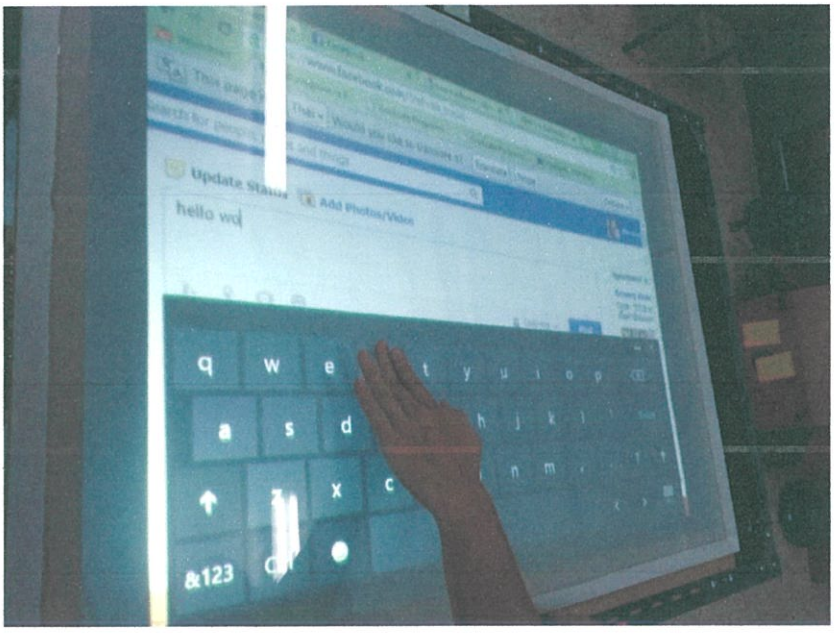


รูปที่ 4.4 การแสดงการพิมพ์บนคีย์บอร์ดหน้าจอสัมผัสได้

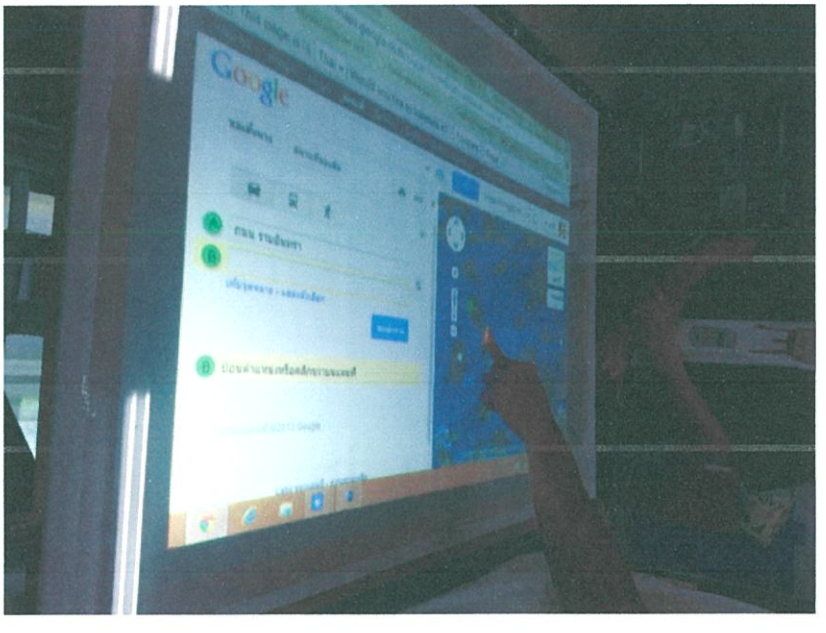
รูปที่ 4.5 การแสดงการใช้งานแทนเม้าส์ โดยใช้งานได้กับทุกโปรแกรมบน Windows 8



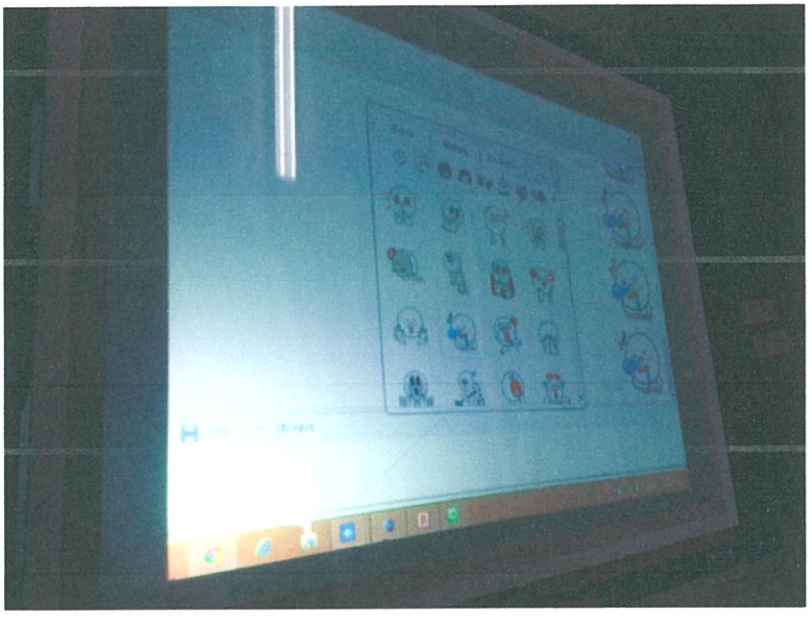
รูปที่ 4.6 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การเล่นเกมส์ออนไลน์



รูปที่ 4.7 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การเล่น Facebook



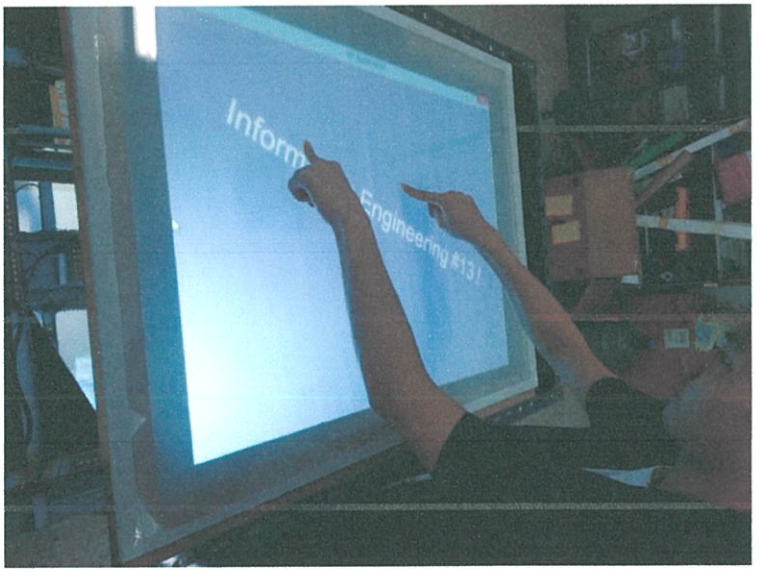
รูปที่ 4.8 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การค้นหาบน Google map



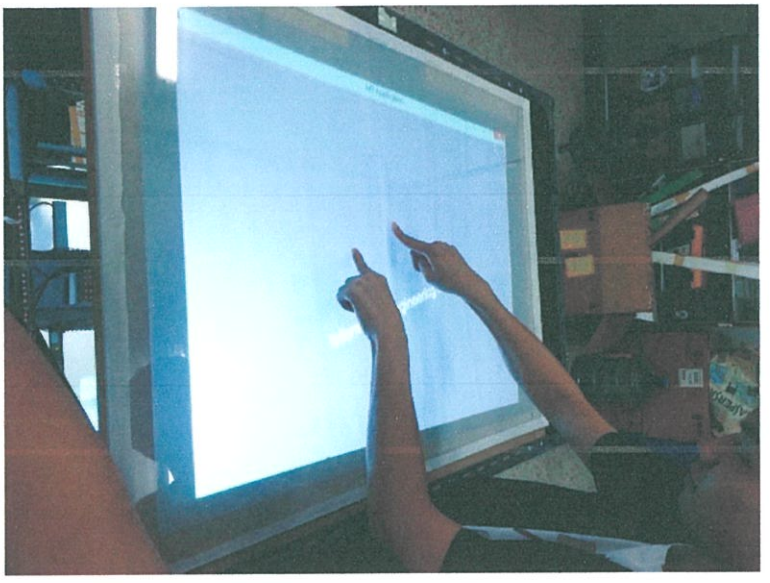
รูปที่ 4.9 การแสดงการใช้งานอินเทอร์เน็ตได้เช่น การใช้โปรแกรมไลน์ (Line)

4.3 ผลการทดลองผลการเชื่อมต่อโปรแกรม CCV กับ แอพพลิเคชั่นที่เขียนด้วยภาษาจาวา

เมื่อทำการตั้งค่าโปรแกรม CCV ให้ส่งข้อมูลการระบุตำแหน่งโดยโปรโตคอล TUIO ไปยังแอพพลิเคชั่นที่เขียนด้วยภาษาจาวา โดยแอพพลิเคชั่นที่เขียนขึ้นรองรับการสัมผัสมากกว่าหนึ่งจุด เช่น โปรแกรมแสดงตัวอักษรที่สามารถย่อและขยายได้โดยการสัมผัส 2 จุดสัมผัสเพื่อย่อและขยายได้ ดังรูปที่ 4.10 และ 4.11



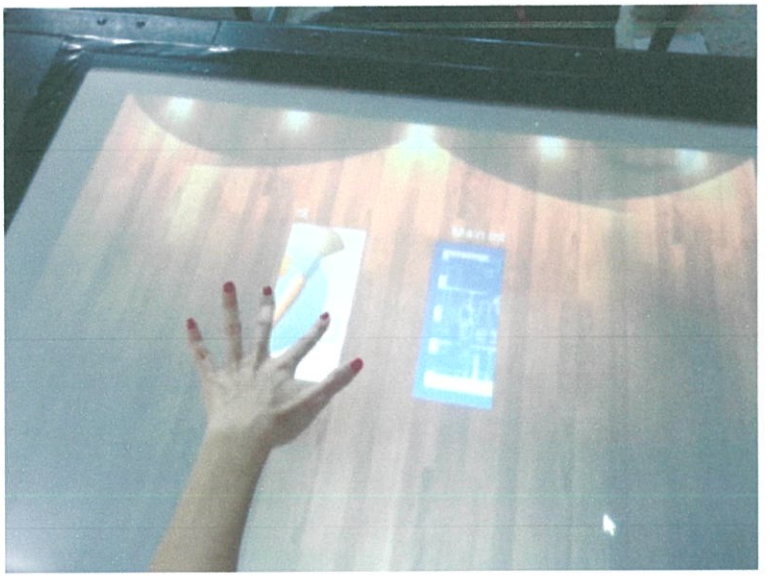
รูปที่ 4.10 การแสดงการสัมผัสขยายตัวอักษรที่เขียนด้วยภาษาจาวา



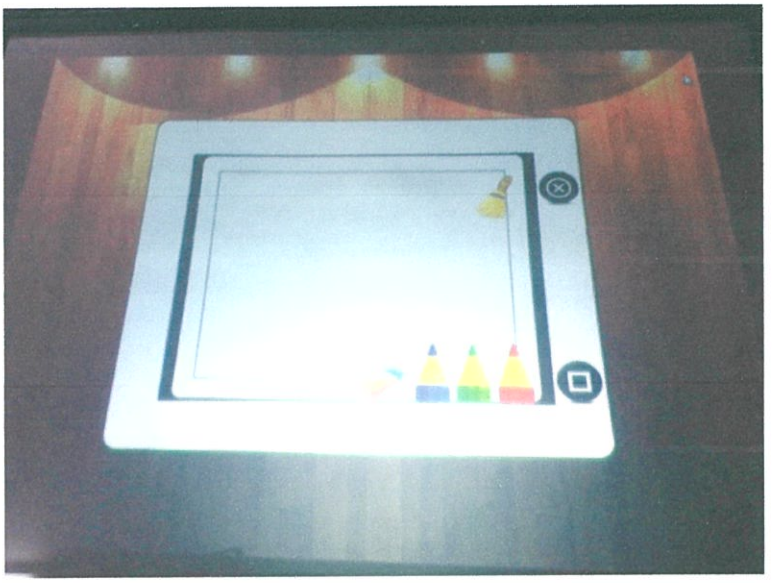
รูปที่ 4.11 การแสดงการสัมผัสย่อตัวอักษรที่เขียนด้วยภาษาจาวา

4.3.1 ผลการทดลองหน้าเมนูหลัก

หน้าจอเมนูหลักจะแสดงแอพพลิเคชั่นที่มีอยู่ (ซึ่งมี 2 แอพพลิเคชั่น) โดยสามารถเลื่อนซ้ายขวาเพื่อเลือกแอพพลิเคชั่นได้ดังรูปที่ 4.12 เมื่อทำการกดเลือกที่ไอคอนแอพพลิเคชั่นนั้นจะแสดงเป็นหน้าต่างเล็กขึ้นมาดังรูปที่ 4.13

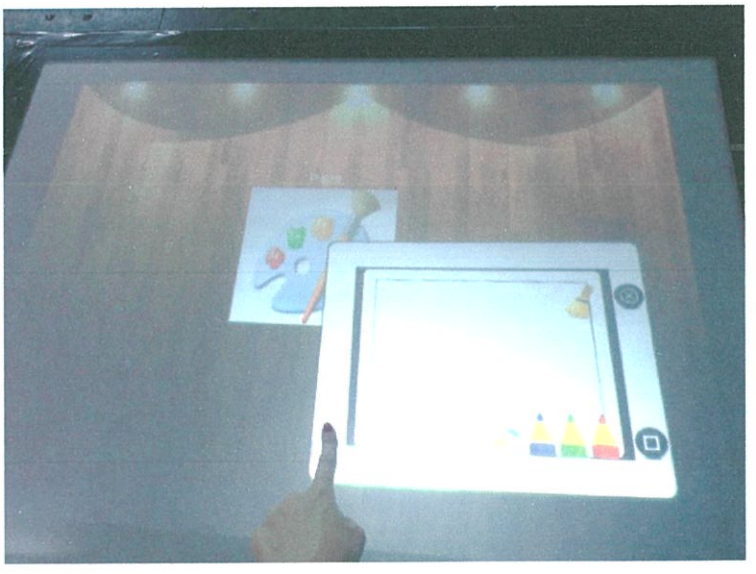


รูปที่ 4.12 การแสดงการเลื่อนซ้ายขวาเพื่อเลือกแอพพลิเคชั่น

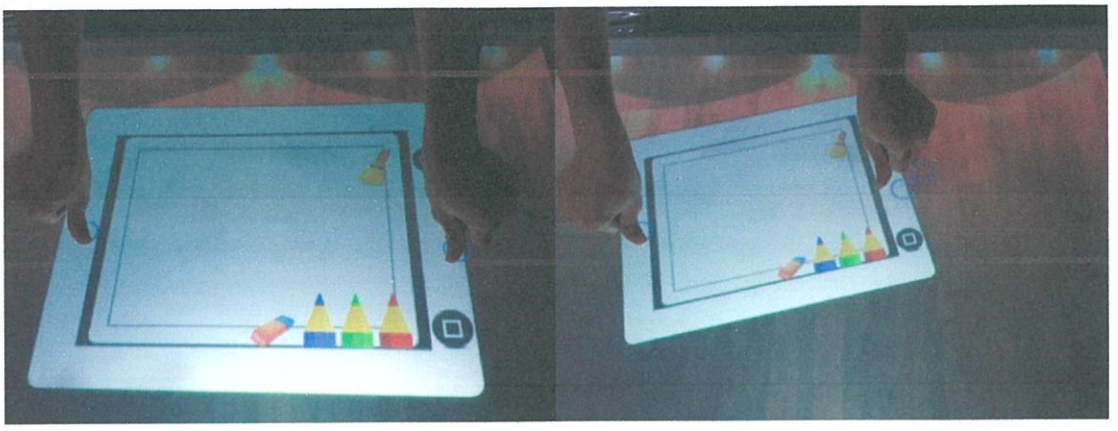


รูปที่ 4.13 การแสดงการเลือกที่ไอคอนแอพพลิเคชั่นนั้นจะแสดงเป็นหน้าต่าง

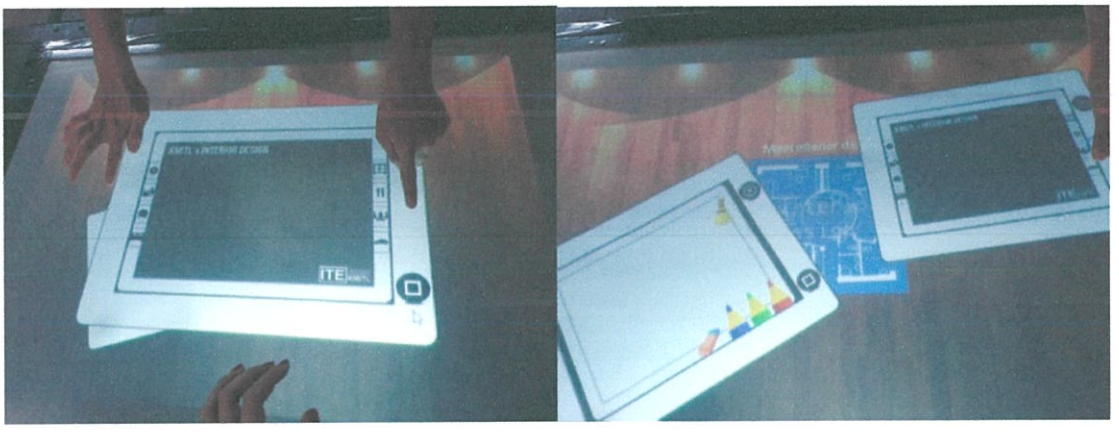
ในส่วนของหน้าต่างแอพพลิเคชั่นสามารถทำการเคลื่อนย้ายและย่อขยายหน้าต่างนี้ได้ ดังรูปที่ 4.14 และ4.15 และสามารถเลือกแอพพลิเคชั่นขึ้นมาอีกได้ดังรูปที่ 4.16 ซึ่งสามารถทำงานได้ทุกแอพพลิเคชั่นที่เลือกขึ้นมา ดังรูปที่ 4.17



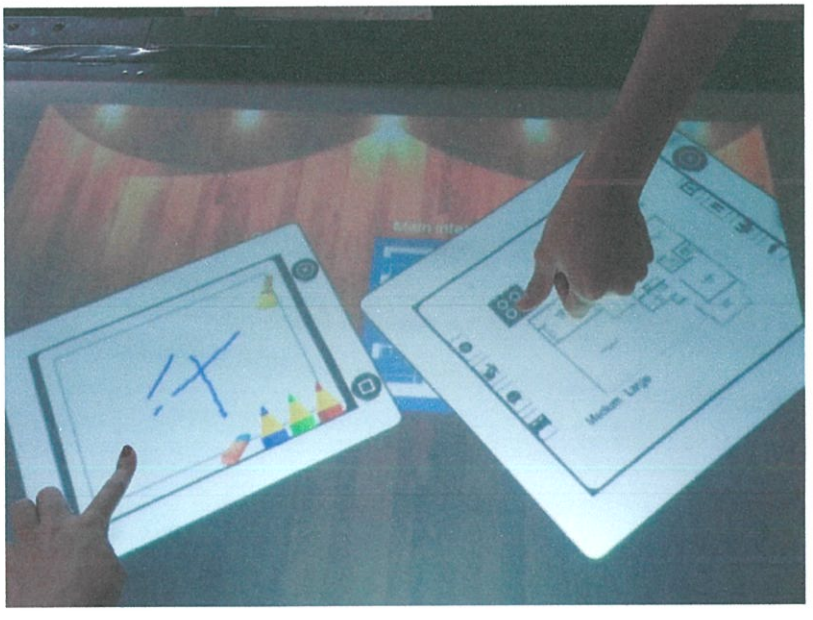
รูปที่ 4.14 หน้าต่างแอพพลิเคชั่นสามารถทำการเคลื่อนย้ายได้



รูปที่ 4.15 การแสดงความสามารถย่อขยายหน้าต่างได้



รูปที่ 4.16 การแสดงความสามารถเลือกแอพพลิเคชั่นขึ้นมาอีกได้

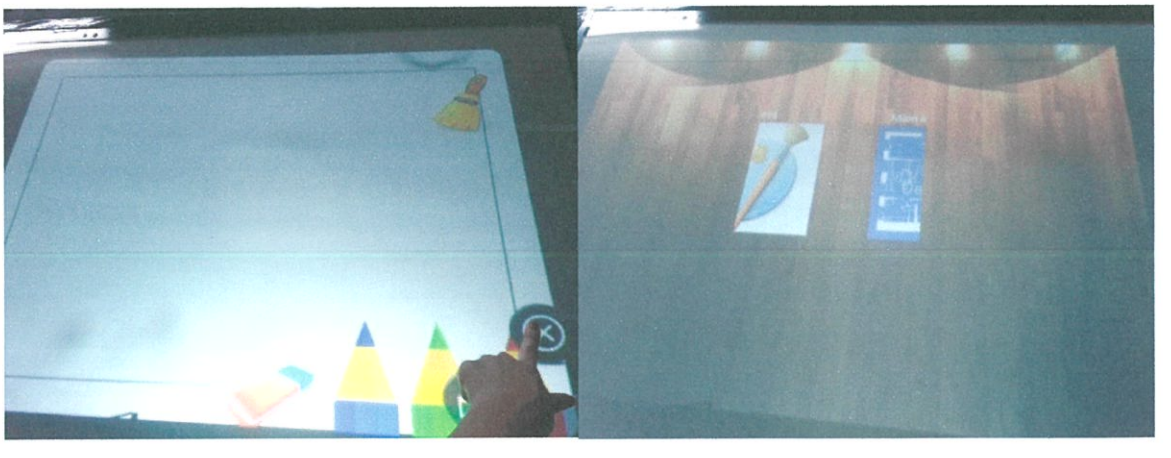


รูปที่ 4.17 การแสดงความสามารถทำงานได้ทุกแอพพลิเคชั่นที่เลือกขึ้นมา

เมื่อต้องการปิดแอพพลิเคชั่นที่เลือกขึ้นมากดปุ่ม X ดังรูปที่ 4.18 หากต้องการขยายแอพพลิเคชั่นที่เลือกให้เต็มหน้าจอกดปุ่มสี่เหลี่ยม ดังรูป 4.19 และหากต้องการปิดโปรแกรมเต็มหน้าจอให้กดตรงมุมขวาแล้วลากมากดปุ่ม X ดังรูปที่ 4.20 จะกลับสู่หน้าเมนูหลัก

รูปที่ 4.18 การปิดแอพพลิเคชั่นที่เลือกขึ้นมากดปุ่ม X

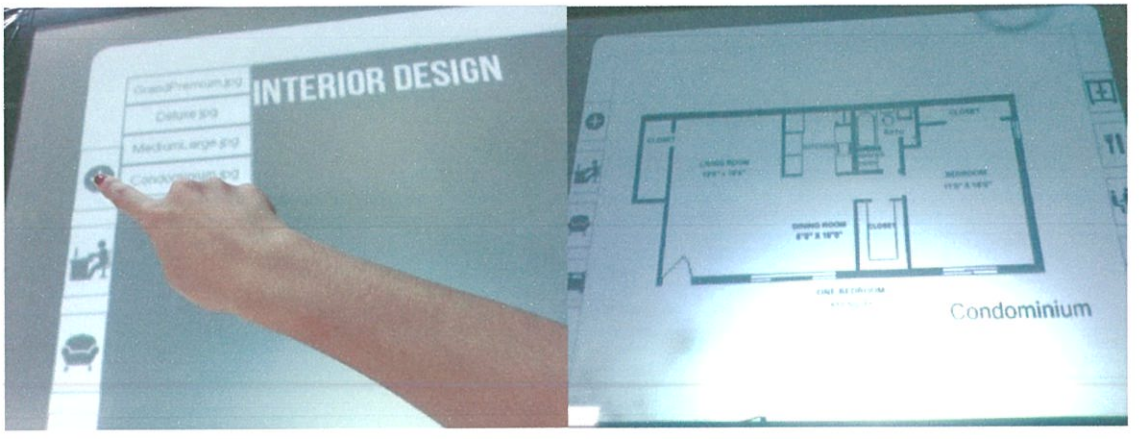
รูปที่ 4.19 การขยายแอพพลิเคชั่นที่เลือกให้เต็มหน้าจอกดปุ่มสี่เหลี่ยม



รูปที่ 4.20 การปิดโปรแกรมเต็มหน้าจอให้กดตรงมุมขวาแล้วลากมากดปุ่ม X จะกลับสู่หน้าเมนูหลัก

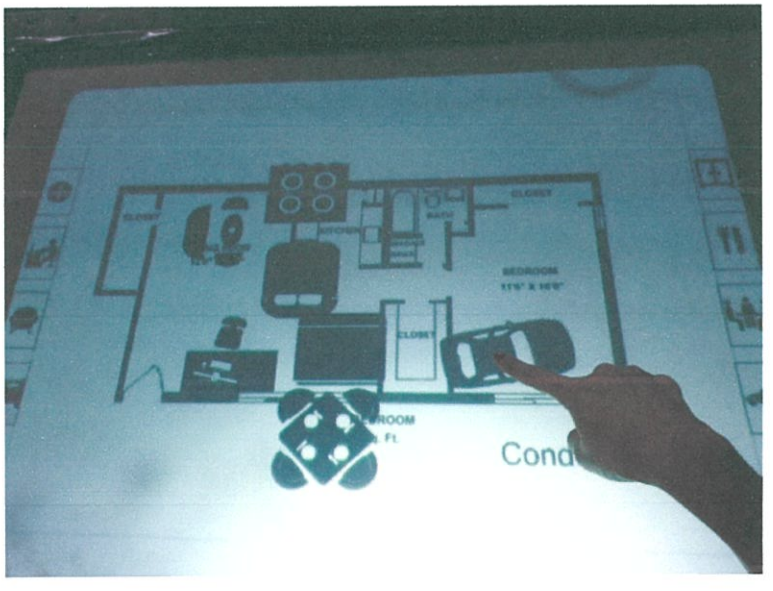
4.3.2 ผลการทดลองแอพพลิเคชั่นออกแบบกตกแต่งภายในบ้าน

หน้าแอพพลิเคชั่นออกแบบกตกแต่งภายในบ้าน สามารถเลือกรูปแบบบ้านดังรูปที่ 4.21

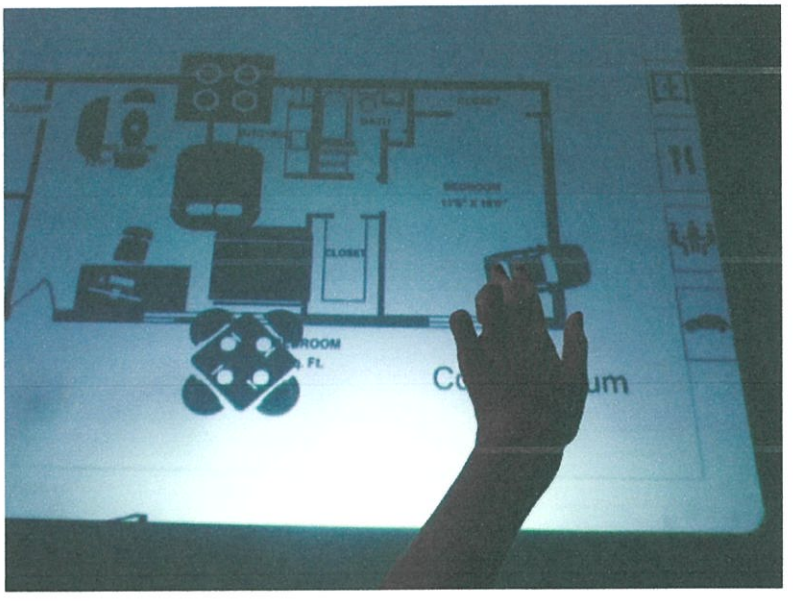


รูปที่ 4.21 การเลือกรูปแบบบ้าน

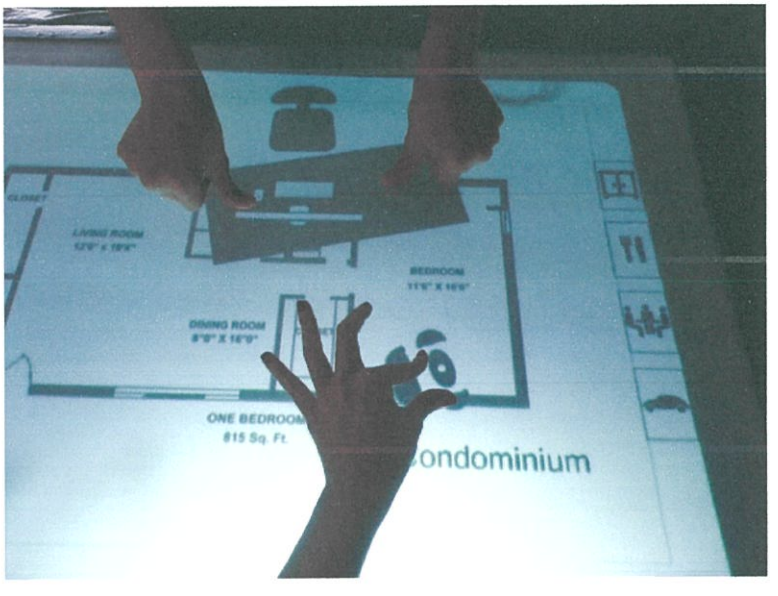
สามารถเลือกเฟอร์นิเจอร์โต๊ะทำงาน โซฟา เตียง ตู้ เตาแก๊ส โต๊ะทานข้าว และรถยนต์ ดังรูปที่ 4.22 ซึ่งเฟอร์นิเจอร์ต่างๆสามารถลากย้ายที่และย่อขยายโดยใช้สองนิ้วกดแล้วลากได้ดังรูปที่ 4.23 และ 4.24



รูปที่ 4.22 การแสดงความสามารถเลือกเฟอร์นิเจอร์โต๊ะทำงาน โซฟา เตียง ตู้ เตาแก๊ส โต๊ะทานข้าว และรถยนต์

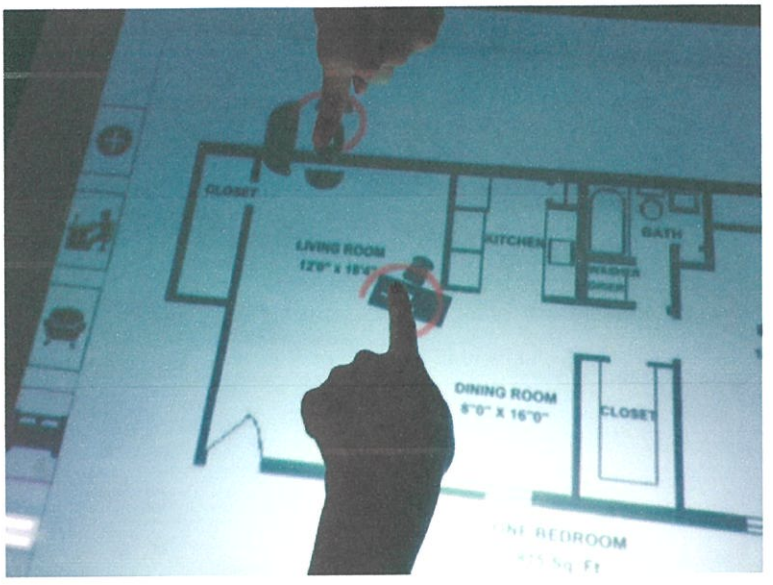


รูปที่ 4.23 เฟอร์นิเจอร์ต่างๆสามารถลากย้ายที่ได้



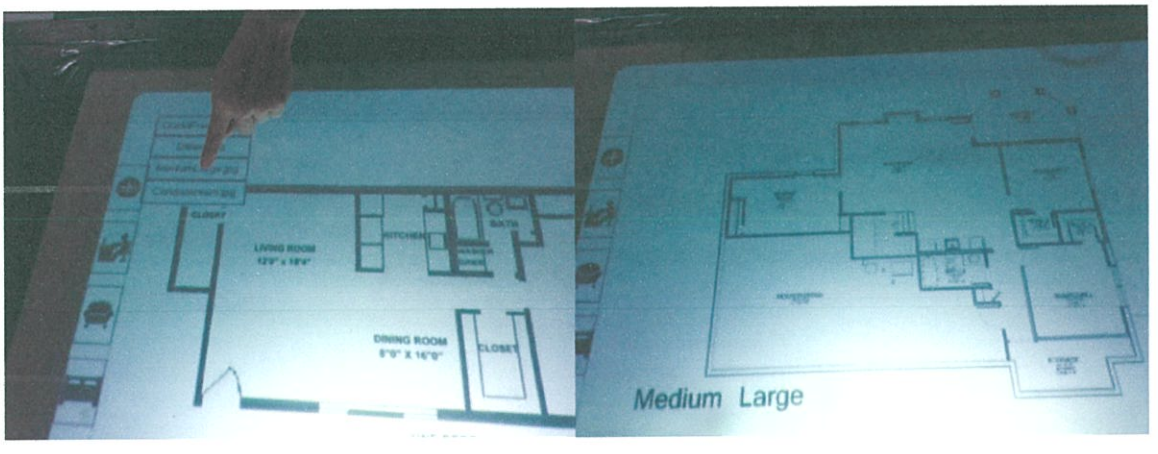
รูปที่ 4.24 ความสามารถในการย่อขยายเฟอร์นิเจอร์ได้โดยใช้สองนิ้วกดแล้วลากได้

เมื่อต้องการลบเฟอร์นิเจอร์ให้กดที่รูปเฟอร์นิเจอร์ที่ต้องการลบค้างไว้จนวงกลมสีแดงวิ่งเต็มวงแล้วรูปเฟอร์นิเจอร์นั้นจะหายไปดังรูปที่ 4.25



รูปที่ 4.25 การลบเฟอร์นิเจอร์ให้กดที่รูปเฟอร์นิเจอร์ที่ต้องการลบค้างไว้จนวงกลมสีแดงวิ่งเต็มวง

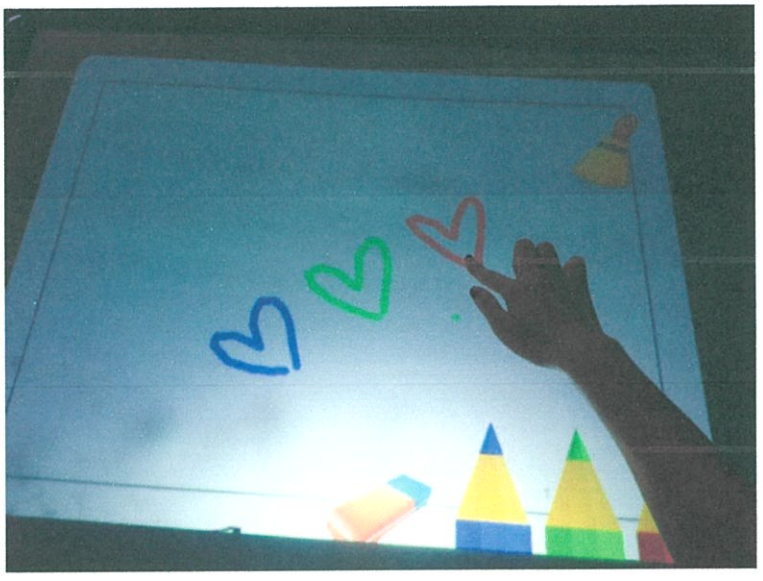
เมื่อต้องการเปลี่ยนรูปแบบบ้าน กด + จะเปลี่ยนแบบบ้านที่แสดงดังรูป 4.26



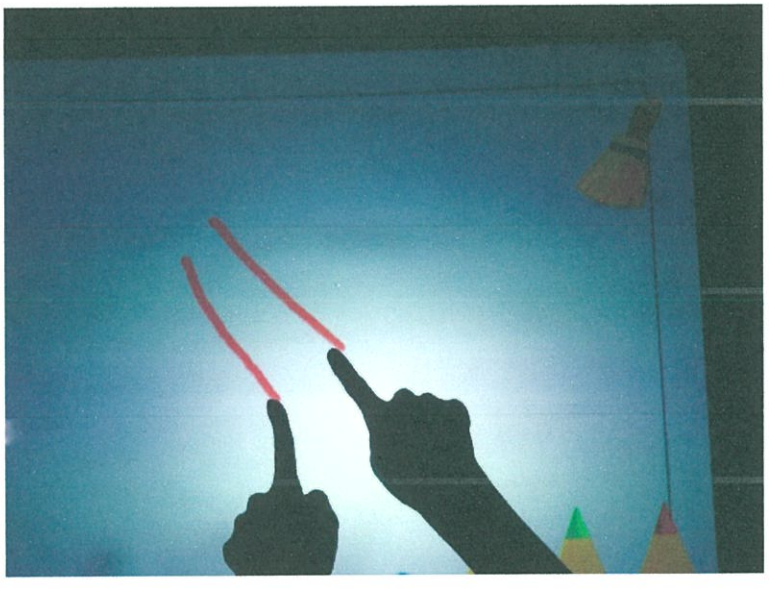
รูปที่ 4.26 เมื่อกด + จะเปลี่ยนแบบบ้านที่แสดง

4.3.3 ผลการทดลองแอพพลิเคชั่นวาดเขียน

แอพพลิเคชั่นวาดเขียนสามารถเขียนได้ 3 สีด้วยกันคือสีน้ำเงิน สีเขียว และสีแดง ดังรูปที่ 4.27 โดยการเขียนนั้นสามารถเขียนได้หลายคนแต่มีข้อจำกัดคือเลือกได้เพียงสีเดียวดังรูปที่ 4.28

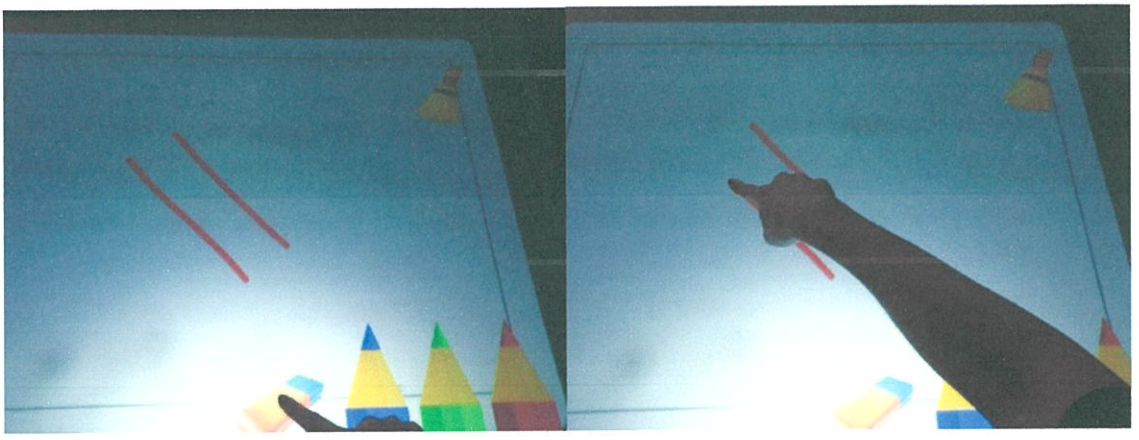


รูปที่ 4.27 การเขียนสามารถเลือกสีของเส้นได้ 3 สีคือสีน้ำเงิน สีเขียว และสีแดง



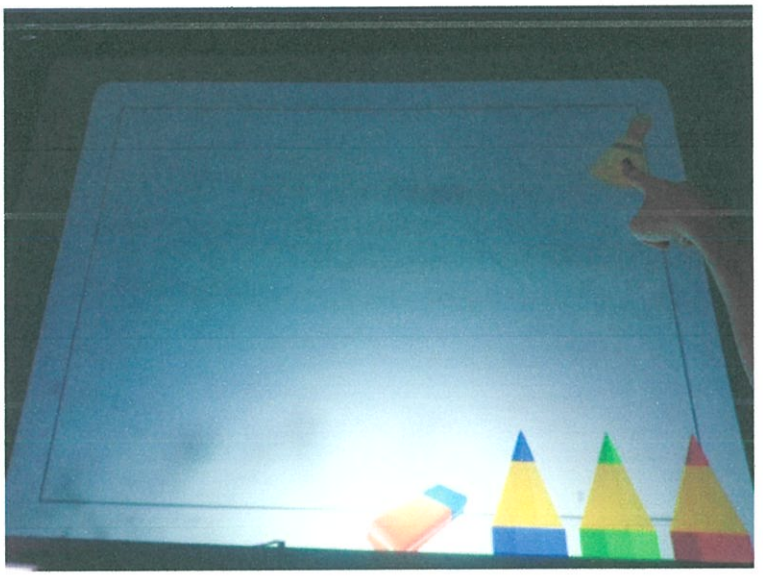
รูปที่ 4.28 ความสามารถเขียนในการเขียนได้หลายคน

เมื่อต้องการลบทีละจุดเลือกที่ยางลบจะเป็นการเขียนด้วยสีขาวดังรูปที่ 4.29



รูปที่ 4.29 การลบทีละจุดเลือกที่ยางลบ แล้วลากไปที่บริเวณที่ต้องการจะลบ

หากต้องการล้างหน้าจอทั้งหมดให้เลือกที่ไม้กวาดจะทำการล้างหน้าจอที่เขียนไปทั้งหมดดังรูปที่ 4.30

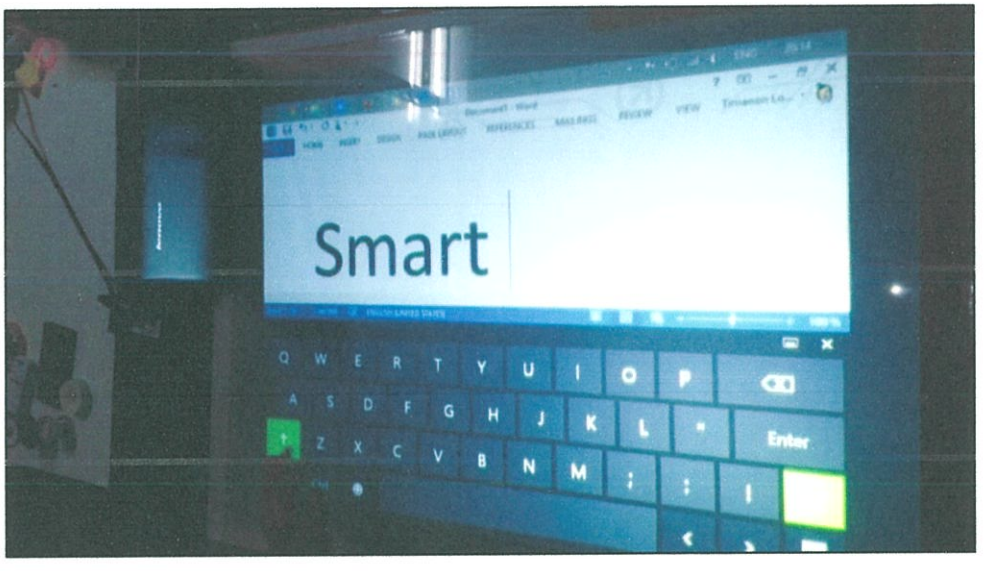


รูปที่ 4.30 การล้างหน้าจอทั้งหมดให้เลือกแปรง

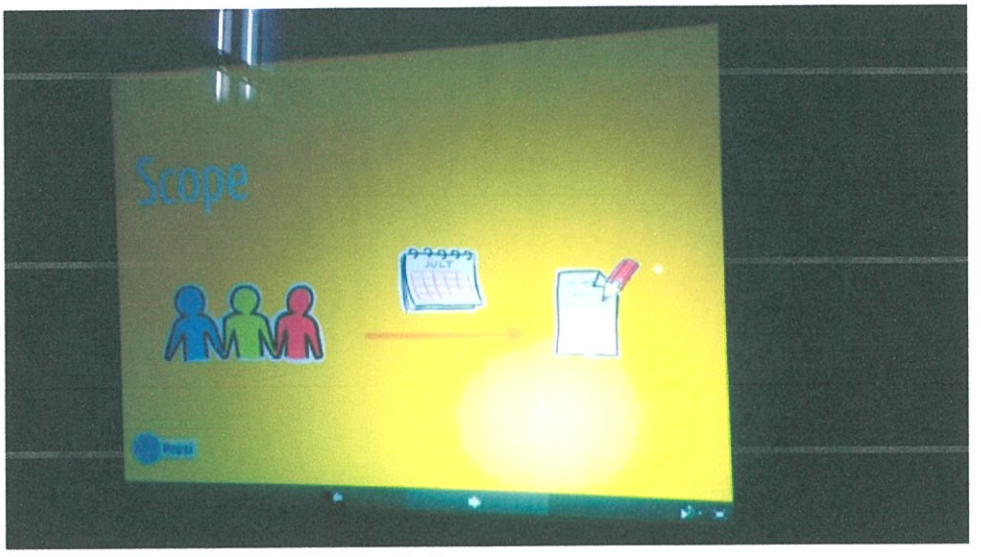
4.4 ผลการทดลองตามขอบเขตที่กำหนด

ขอบเขตที่ 1 สามารถเป็นหน้าจอสัมผัสของคอมพิวเตอร์ได้ โดยการใช้เทคโนโลยีหน้าจอสัมผัสอินฟราเรดเทคนิค Rear Diffused Illumination (Rear DI) ทำให้ผู้ใช้สามารถใช้งานโต๊ะอัจฉริยะเป็นหน้าจอสัมผัสของคอมพิวเตอร์ได้

ขอบเขตที่ 2 สามารถใช้งานกับ Microsoft Windows 8 โดยใช้การสัมผัสเพื่อป้อนอินพุตต่างๆ ได้ โดยการใช้โปรแกรม Touch Injector ในการรับค่าการระบุตำแหน่งนิ้วมือและเชื่อมต่อไปยัง Microsoft Windows 8 ทำให้ผู้ใช้สามารถใช้งาน Windows 8 โดยการสัมผัสหน้าจอได้ สามารถใช้โปรแกรม Microsoft Word และใช้คีย์บอร์ดของ Microsoft Windows 8 ได้ ดังรูปที่ 4.31 และ สามารถใช้ร่วมกับโปรมแกรมต่างๆ เช่นโปรแกรมนำเสนอผลงาน Prezi ดังรูปที่ 4.32 ได้

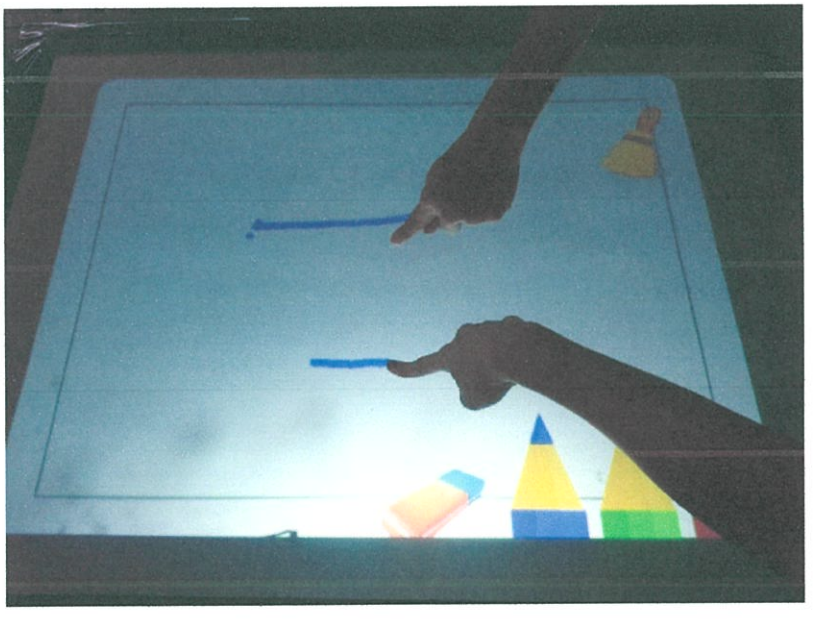


รูปที่ 4.31 การใช้ทัชคีย์บอร์ดกับโต๊ะอัจฉริยะ

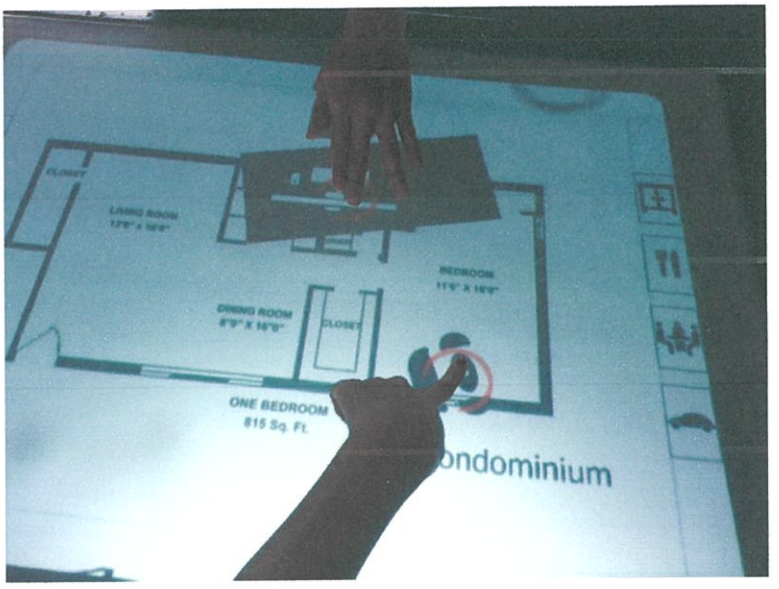


รูปที่ 4.32 การใช้โต๊ะอัจฉริยะในการนำเสนอผลงาน

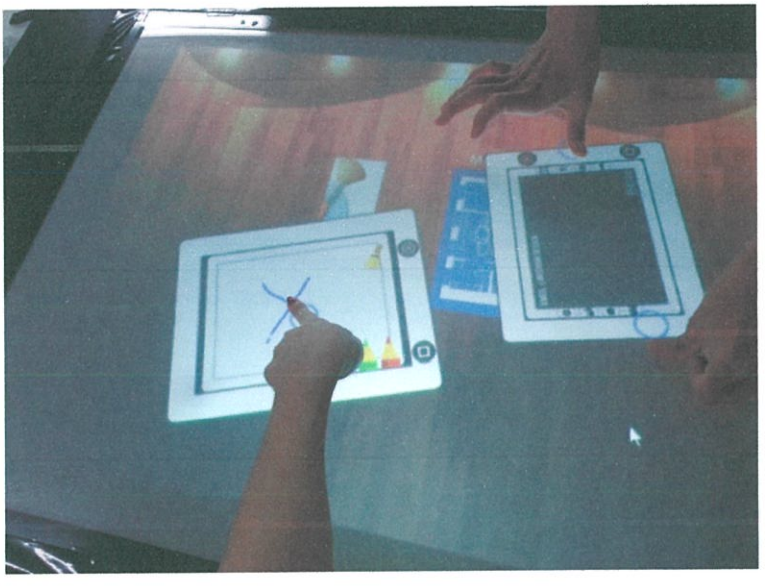
ขอบเขตที่ 3 สามารถรองรับผู้ใช้งานได้มากกว่าหนึ่งคนเนื่องจากเทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรดเป็นเทคโนโลยีที่รองรับการสัมผัสได้มากกว่าหนึ่งจุดทำให้สามารถมีผู้ใช้ได้มากกว่าหนึ่งคน ซึ่งขึ้นอยู่กับแอพพลิเคชั่นที่เลือกใช้ โดยส่วนของแอ๊พพลิเคชั่นที่สามารถทำงานร่วมกับ Microsoft Windows8 นั้นไม่สามารถรองรับการสัมผัสมากกว่าหนึ่งจุดได้ จึงได้เขียนแอพพลิเคชั่นโดยใช้ภาษาจาวาเพื่อรองรับการสัมผัสได้มากกว่าหนึ่งจุดได้ดังรูปที่ 4.33 รูปที่ 4.34 และรูปที่ 4.35



รูปที่ 4.33 แอพพลิเคชั่นที่สามารถรองรับการสัมผัสได้มากกว่าหนึ่งจุด

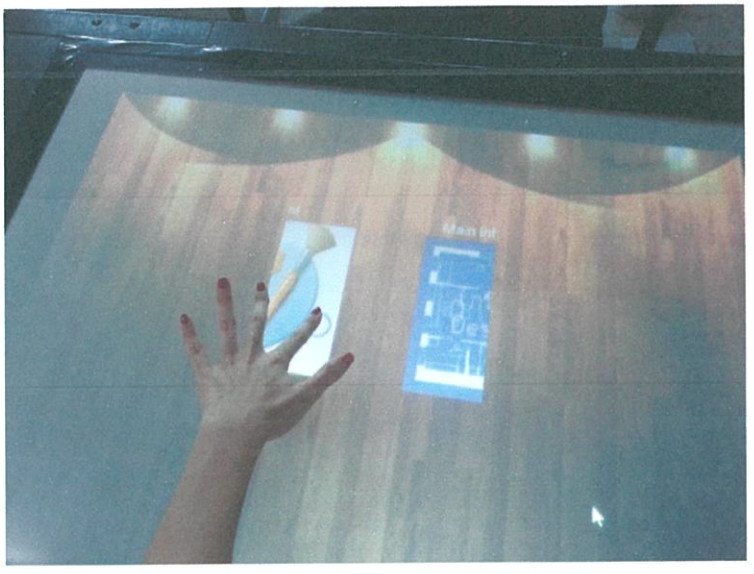


รูปที่ 4.34 แอพพลิเคชั่นที่สามารถรองรับการสัมผัสได้มากกว่าหนึ่งจุด

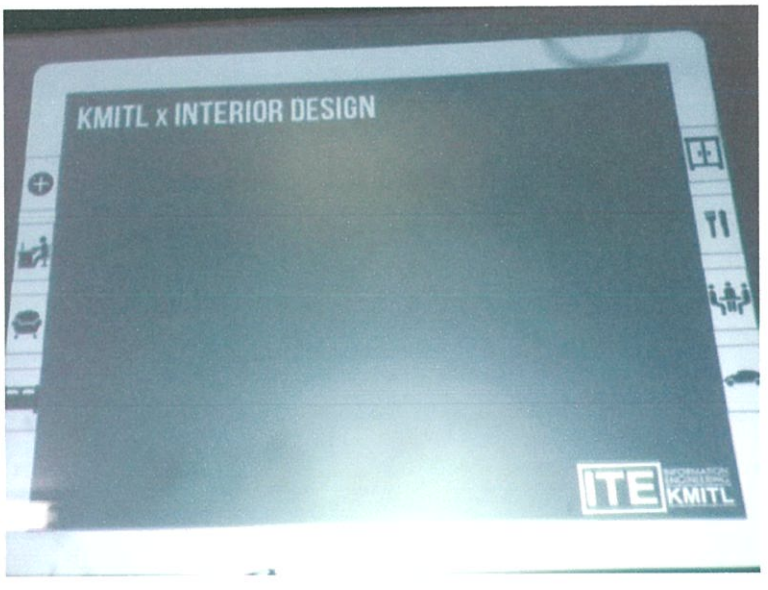


รูปที่ 4.35 แอพพลิเคชั่นที่สามารถรองรับการสัมผัสได้มากกว่าหนึ่งจุด

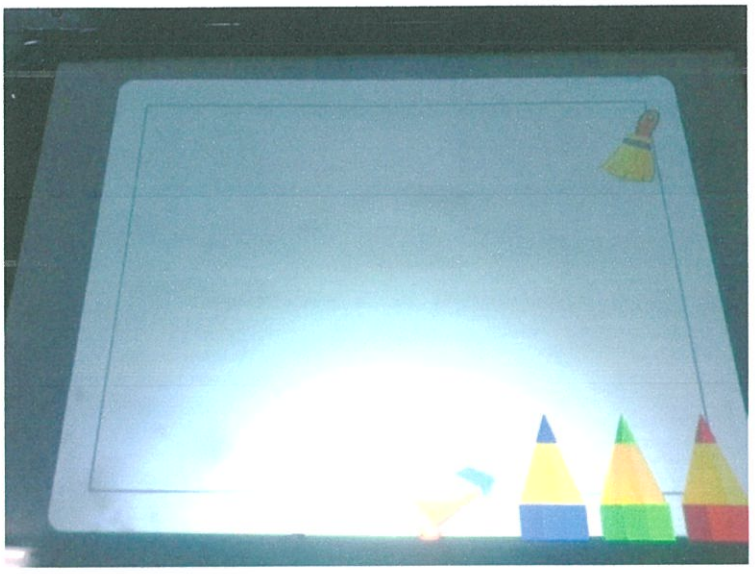
ขอบเขตที่ 4 สามารถใช้งานแอพพลิเคชั่นเพิ่มเติม โดยได้เขียน หน้าจอเมนูหลักเพื่อมารวบรวมแอพพลิเคชั่นดังรูปที่ 4.36 และได้เขียนแอพพลิเคชั่นด้วยภาษาจาวา 2 แอพพลิเคชั่นด้วยกันคือ แอพพลิเคชั่นออกแบบกตกแต่งภายในบ้าน ดังรูปที่ 4.37 และ แอพพลิเคชั่นวาดเขียน ดังรูปที่ 3.38



รูปที่ 4.36 หน้าจอเมนูหลัก



รูปที่ 4.37 แอพพลิเคชั่นออกแบบกตกแต่งภายในบ้าน



รูปที่ 4.38 แอพพลิเคชั่นวาดเขียน

บทที่ 5
สรุปผลและแนวทางการแก้ไข

5.1 สรุปผลการทำงาน

จากการศึกษาการทำงานของเทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรด และการศึกษาโปรแกรมของซอฟต์แวร์ที่ที่ใช้ในการทำงาน สามารถสร้างหน้าจอสัมผัสที่สามารถใช้งานที่หลากหลายได้ โดยหน้าจอสัมผัสที่สร้างขึ้นนี้สามารถเป็นหน้าจอสัมผัสของคอมพิวเตอร์ และสามารถใช้งานร่วมกับ Microsoft Windows 8 ได้ และเนื่องจากเทคโนโลยีหน้าจอสัมผัสแบบอินฟราเรดเป็นเทคโนโลยีที่รองรับการสัมผัสได้มากกว่าหนึ่งจุดทำให้สามารถมีผู้ใช้ได้มากกว่าหนึ่งคน แต่ขึ้นอยู่กับการรองรับของแอพพลิเคชั่น และ Microsoft Windows 8 รองรับการสัมผัสเพียงหนึ่งจุด จึงได้เขียนแอพพลิเคชั่นที่รองรับการสัมผัสมากกว่าหนึ่งจุดโดยใช้ภาษาจาวาในการสื่อสารผ่านโปรโตคอล TUIO กับโปรแกรม CCV เพื่อรับจุดสัมผัสโดยตรงทำให้สามารถสัมผัสมากกว่าหนึ่งจุดได้

5.2 ปัญหาระหว่างการดำเนินงาน

- แอลอีดีอินฟราเรดมีคุณสมบัติที่มีประสิทธิภาพต่ำกว่าแอลอีดีอินฟราเรดที่ควรใช้ เนื่องจากแอลอีดีอินฟราเรดที่ควรใช้ควรมีองศาการฉายแสงที่ 45 องศา แต่แอลอีดีอินฟราเรดที่มีขายในประเทศไทยนั้นมีองศาเพียง 20 องศา ทำให้การฉายแสงอินฟราเรดนั้นมีประสิทธิภาพที่น้อยกว่า และทำให้ต้องใช้แอดอีดีอินฟราเรดจำนวนมาก
- การจะทำให้แสงอินฟราเรดกระจายเพียงพอต่อการใช้เทคโนโลยีหน้าจอสัมผัสเทคนิค Rear Diffused Illumination ทำได้ยาก
- การสัมผัสหน้าจอหากมีส่วนแขนหรืออุ้งมือสัมผัสหน้าจอจะทำให้เกิดการระบุตำแหน่งของนิ้วมือที่ผิดพลาด ซึ่งสามารถตั้งค่าการระบุตำแหน่งได้ยาก
- เครื่องฉายภาพมีระยะการฉายภาพที่มากกว่าความสูงของโต๊ะอยู่มาก
- กล้องเว็บแคมเพียงตัวเดียวไม่สามารถจับภาพทั้งหน้าจอได้

5.3 แนวทางการแก้ไข

- ออกแบบโดยการวางแอลอีดีเป็นแผงยาวรอบๆและฉายขึ้นไปยังหน้าจอเพื่อให้แสงไปถึงหน้าจอได้ง่าย และมีการวางหลอดแอลอีดีหน้ากระจกเงาเพื่อสะท้อนขึ้นไปยังหน้าจอตรงที่แอลอีดีที่วางรอบๆนั้นส่องไม่ถึง เพื่อให้มีแสงอินฟราเรดที่เพียงพอต่อการใช้งาน
- ทำการตั้งค่าการระบุตำแหน่งโดยกำหนด Min Blob Size และ Max Blob Size เรื่องจากการสัมผัสที่ไม่ใช่นิ้วมือจะมีขนาดของจุดสัมผัสต่างกับจุดสัมผัสที่เกิดจากนิ้วมือ
- ทำการสะท้อนภาพที่ได้จากเครื่องฉายภาพโดยใช้กระจกเงา 2 บานสะท้อนขึ้นไปยังหน้าจอ
- ทำการเพิ่มกล้องขึ้นมากอีกหนึ่งตัว โดยจับภาพคนละครึ่งของหน้าจอสัมผัสและทำการสะท้อนหนึ่งครั้งเพื่อให้จับภาพได้เต็มหน้าจอ

5.4 แนวทางการพัฒนาต่อ

- พัฒนาแอพพลิเคชั่นเพิ่มเติมทำให้สามารถใช้งานได้หลากหลายมากยิ่งขึ้น
- เพิ่มประสิทธิภาพในการสัมผัสให้มีความแม่นยำมากยิ่งขึ้นโดยอาจจะเพิ่มแสงอินฟราเรดหรือเพิ่มการสะท้อนแสงอินฟราเรด

บรรณานุกรม

[1] http://wiki.nuigroup.com/Diffused_Illumination

[2] http://sethsandler.com/multitouch/ftir/

[3] http://sethsandler.com/multitouch/llp/

[4] http://sethsandler.com/multitouch/dsi/

[5] http://sethsandler.com/multitouch/ledlp/

[6] http://sethsandler.com/multitouch/frontdi/

[7] http://sethsandler.com/multitouch/reardi/

[8] http://www.thai-av.com/index.php?lay=show&ac=article&Id=539016614&Ntype=2

[9] http://www.maceducation.com/e-knowledge/2432209100/18.htm

[10] http://www.rmutphysics.com/physics/oldfront/62/light1/ligh_4.htm

[11] http://www.tuio.org/

[12] http://www.mindphp.com/

[13] http://www.logitech.com/th-th/product/webcam-c170

[14] http://wiki.nuigroup.com/How_to_Modify_Cam

[15] http://www.projectormba.com/Projector-EPSON/Epson-Smart-Series/Epson-Business-EB-S02.html

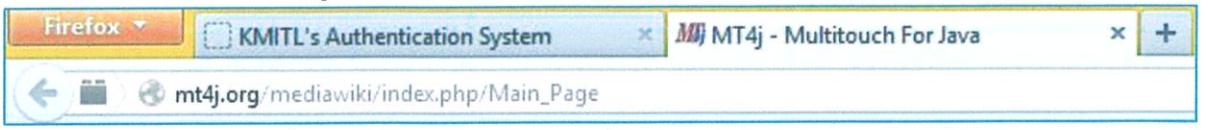
[16] https://attachment.fbsbx.com/file_download.php?id=502931946446071

ภาคผนวก

คู่มือการติดตั้งโปรแกรมสำหรับโต๊ะอัจฉริยะ

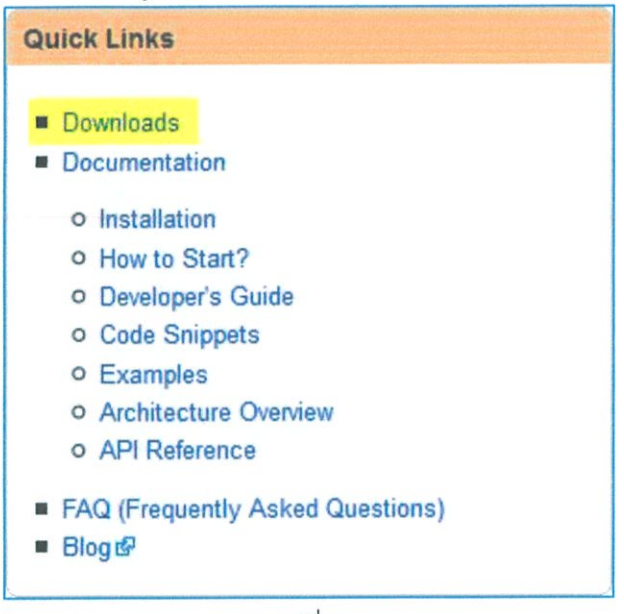
ขั้นตอนการติดตั้ง framework MT4j

1. เปิดเว็บไซต์ www.mt4j.org ตามรูปที่ 1



รูปที่ 1

2. เลือกเมนู Downloads ทางด้านขวามือ ตามรูปที่ 2



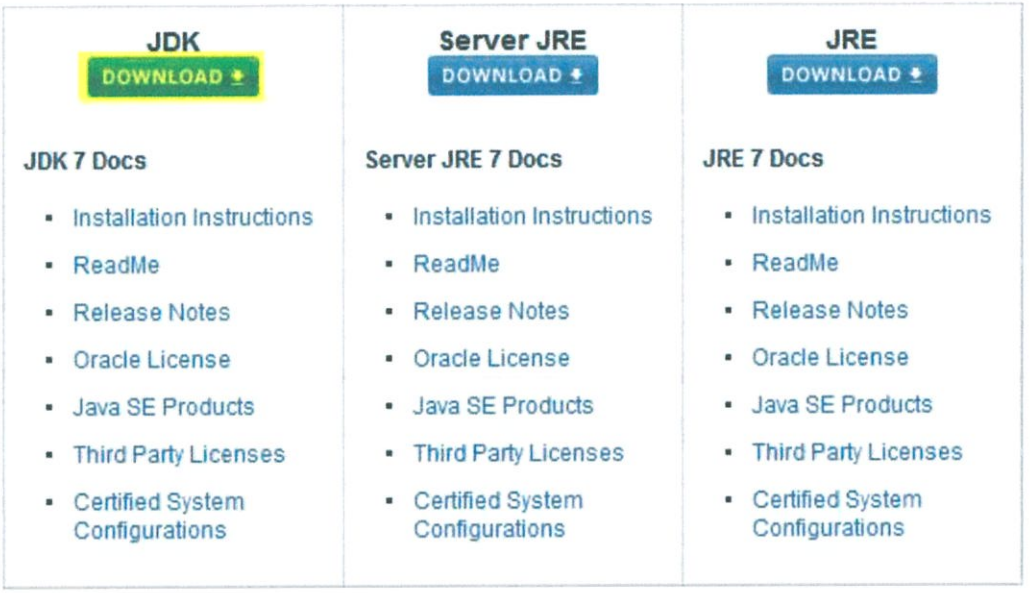
รูปที่ 2

3. จะมี framework ให้เลือกดาว์โหลดอยู่ 3 ตัวโดย
   - MT4j version 0.98 full (dev. preview) release คือ framework ที่มี example project รวมมาไว้ใน framework ด้วย
   - MT4j version 0.98 binary (dev. preview) release คือ framework ที่แต่
   - MT4j Android alpha (dev. preview) release คือ framework ที่ไว้ใช้พัฒนากับ Android

   ให้เลือกดาว์โหลด MT4j version 0.98 full ตามรูปที่ 3

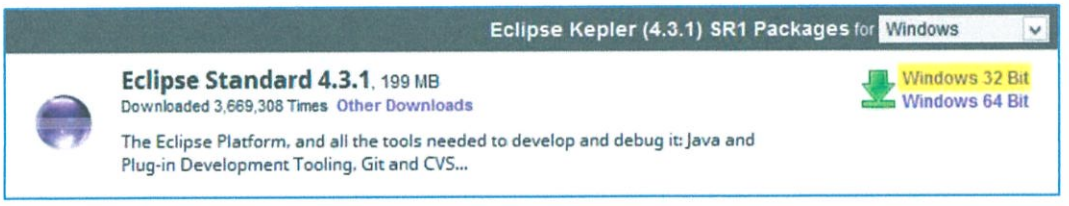
รูปที่ 3

4. เนื่องจาก framework MT4j จะทำงานกับ Java™ 6 Development Kit เวอร์ชั่น 32 bit ซึ่งหาดาว์โหลดได้จากเว็บไซต์ http://www.oracle.com/technetwork/java/javase/downloads/index.html ตามรูปที่ 4



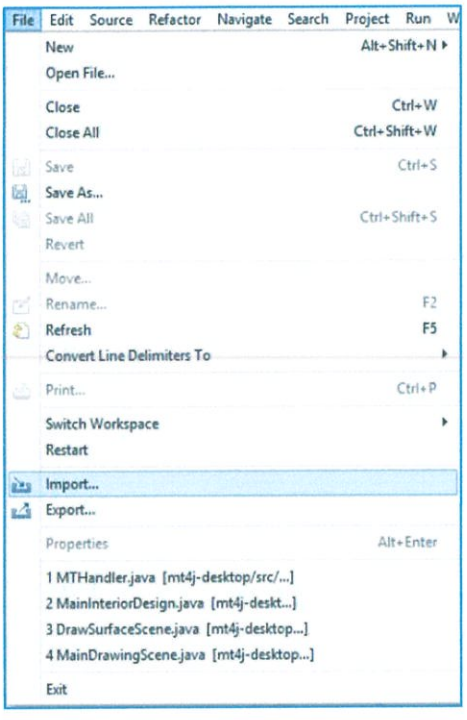
รูปที่ 4

5. ดาว์โหลด eclipse โดยเลือกเวอร์ชั่น 32 bit จากเว็บไซต์ http://eclipse.org/downloads/ ตามรูปที่ 5



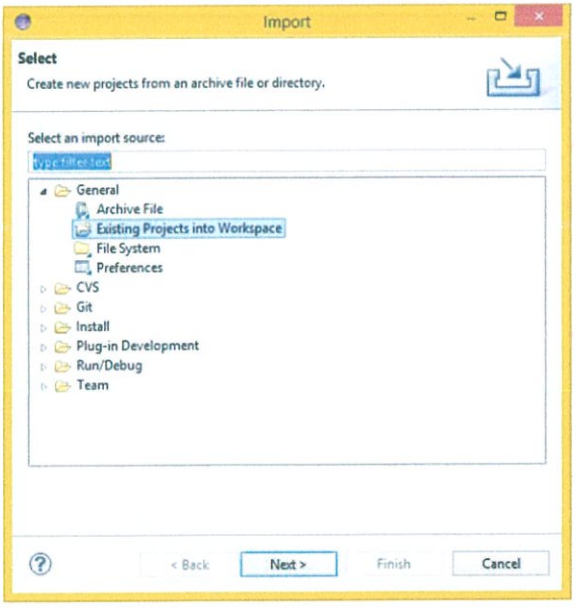
รูปที่ 5

6. เปิด eclipse เลือก File -> Import ตามรูปที่ 6



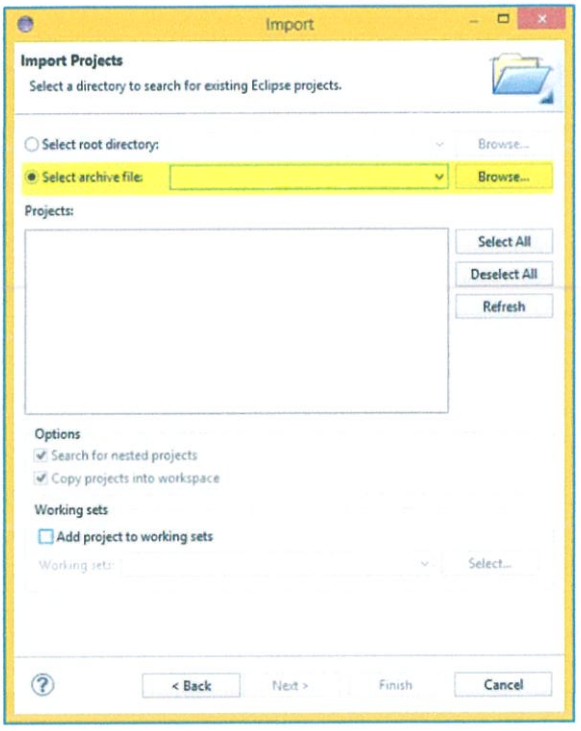
รูปที่ 6

7. เลือก Import existing project into workspace ตามรูปที่ 7



รูปที่ 7

8. เลือก Select archive file แล้วกด Browse... ตามรูปที่ 8



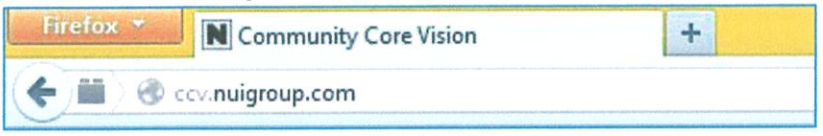
รูปที่ 8

9. Browse ไปยังโฟลเดอร์ที่ดาว์โหลด mt4xxx.zip ไว้แล้วกด Open

10. กด Finish

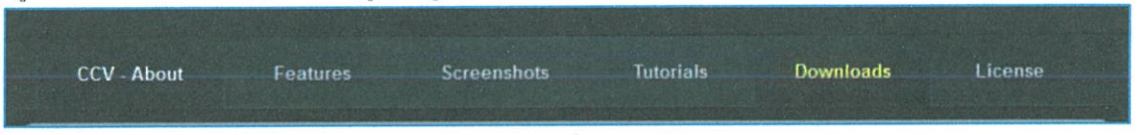
ขั้นตอนการติดตั้งโปรแกรม CCV

1. เปิดเว็บไซต์ www.ccv.nuigroup.com ตามรูปที่ 1



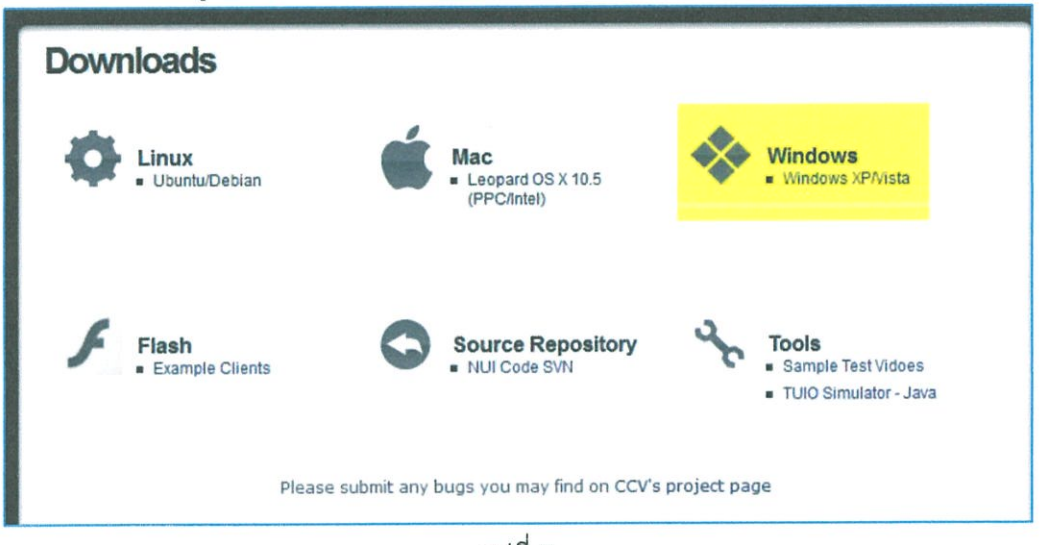
รูปที่ 1

2. เลือกเมนู Download ทางด้านบนของเมนู ตามรูปที่ 2



รูปที่ 2

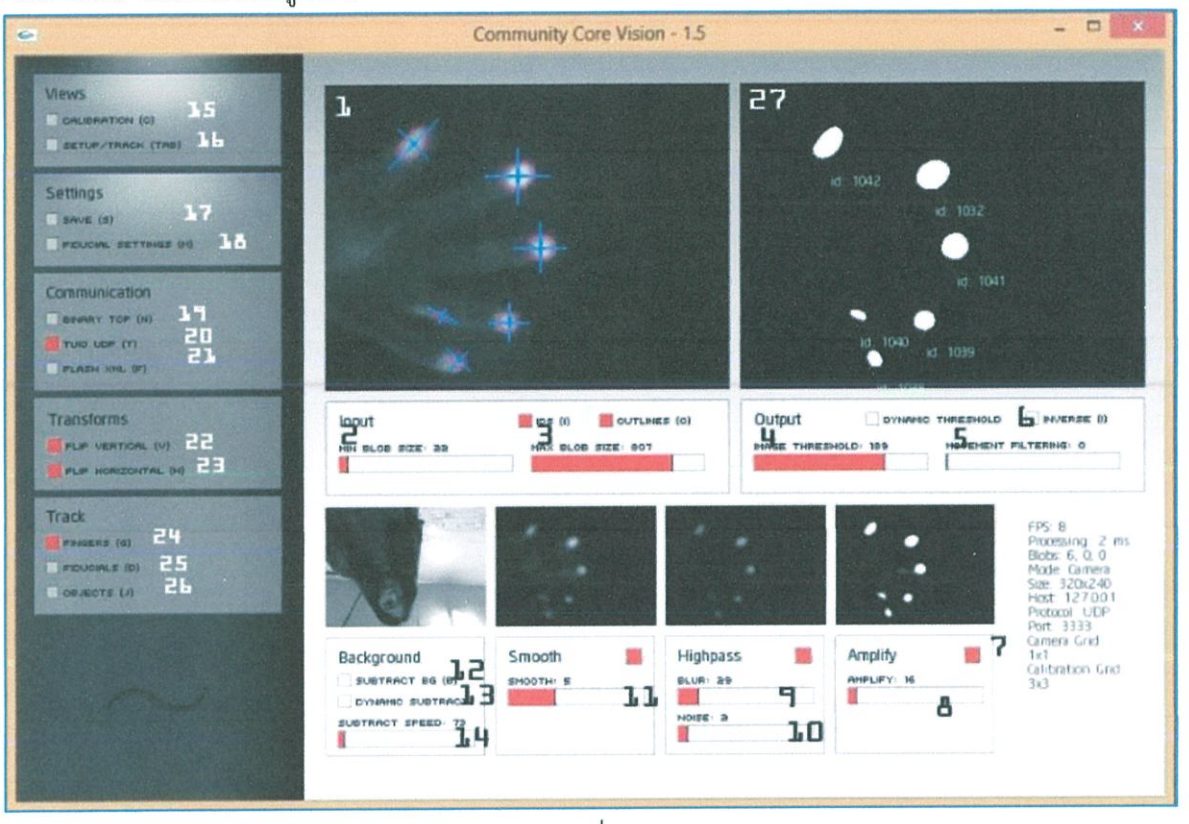
3. เลือกประเภท Windows ตามรูปที่ 3



รูปที่ 3

4. จากนั้นทำการติดตั้งโปรแกรมให้เรียบร้อย

5. Interface โปรแกรมมีดังนี้ ตามรูปที่ 4



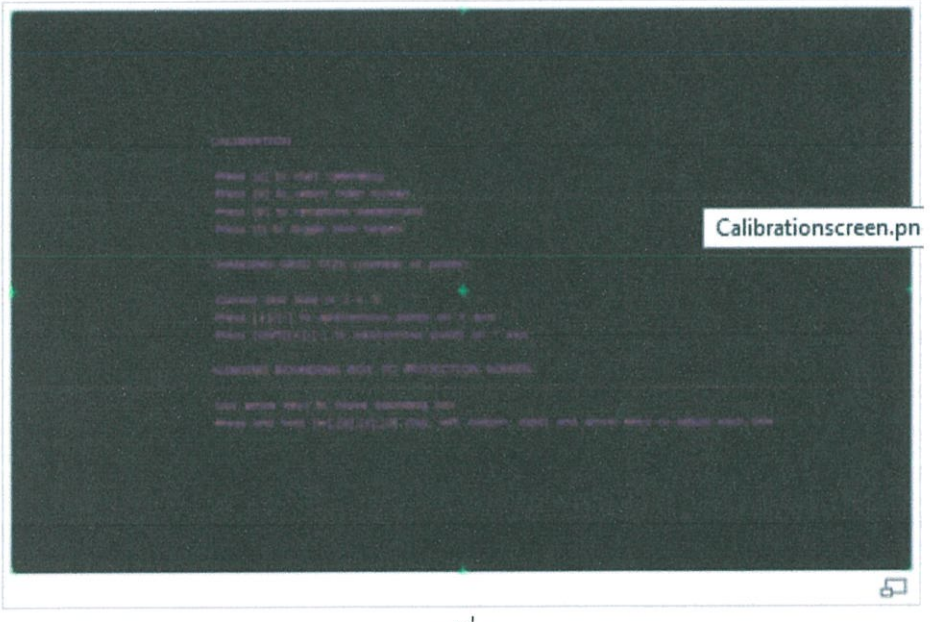
รูปที่ 4

***หมายเหตุ**

1. Source image: แสดงวีดีโอจากกล้องหรือไฟล์วีดีโอ
2. Min blob size: กำหนดจุดสัมผัสที่เล็กที่สุดที่ต้องการ หากมีจุดเล็กกว่าที่กำหนดจะไม่ทำการระบุตำแหน่ง
3. Max Blob size: กำหนดจุดสัมผัสที่ใหญ่ที่สุดที่ต้องการ หากมีจุดที่ใหญ่กว่าที่กำหนดจะไม่ทำการระบุตำแหน่ง
4. Threshold slider: ปรับระดับของพิกเซลของการระบุตำแหน่งที่สามารถยอมรับได้
5. Movement filtering: ปรับระดับของระยะทาง ก่อนที่จะมีการตรวจพบจุดสัมผัส
6. Inverse: เปลี่ยนจุดสัมผัสสีดำเป็นสีขาว
7. On/off toggle: เปิด/ปิด การใช้งานฟังก์ชั่นต่างๆ
8. Amplify slider: ทำให้จุดตำแหน่งสัมผัสที่อ่อน ชัดขึ้น
9. Highpass blur slider: นำส่วนที่เบลอออก และนำส่วนที่ชัดให้มีความคมชัดขึ้น
10. Highpass noise: กรองในส่วนที่เป็น noise ออก
11. Smooth slider: ปรับความเรียบเนียนของภาพ
12. Remove background button: ลบภาพต่างๆออกไป เหลือเพียงภาพเริ่มต้น
13. Dynamic subtract toggle: กำหนดการเปลี่ยนแปลงของสิ่งแวดล้อมที่เป็นพื้นหลัง เพื่อไม่ให้รบกวนระบบ

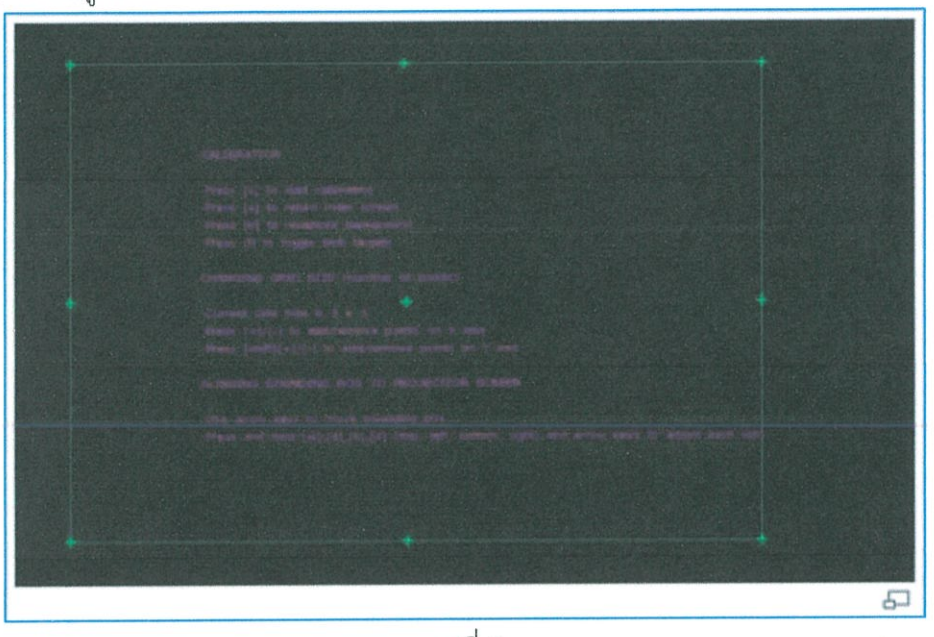
14. Subtrat speed: ปรับค่าการเปลี่ยนแปลงของพื้นหลังเพื่อไม่ให้รบกวนระบบ
15. Calibration: การปรับตั้งค่าหน้าจอให้ตรงกับหน้าจอที่ใช้งานจริง
16. Setup/Track: การกำหนดอินพุต และการตั้งค่ากล้องที่ใช้
17. Save: บันทึกค่า
18. Fiducial setting: การตั้งค่าความแม่นยำ
19. Communication by Binary TCP: การส่งข้อมูลโดยใช้ Binary TCP
20. Communication by TUIO UDP: การส่งข้อมูลโดยใช้ TUIO UDP
21. Communication by FLASH XML: การส่งข้อมูลโดยใช้ FLASH XML
22. Flip vertical: สลับด้านแนวตั้ง
23. Flip horizontal: สลับด้านแนวนอน
24. Track fingers: ทำการระบุตำแหน่งนิ้วมือ
25. Track fiducials: ทำการระบุตำแหน่งที่ต้องการความแม่นยำ
26. Track objects: ทำการระบุตำแหน่งวัตถุทั่วไป

6. การ Calibration ทำได้โดยการกดอักษร “C” บนแป้นคีย์บอร์ดจะได้ ตามรูปที่ 5



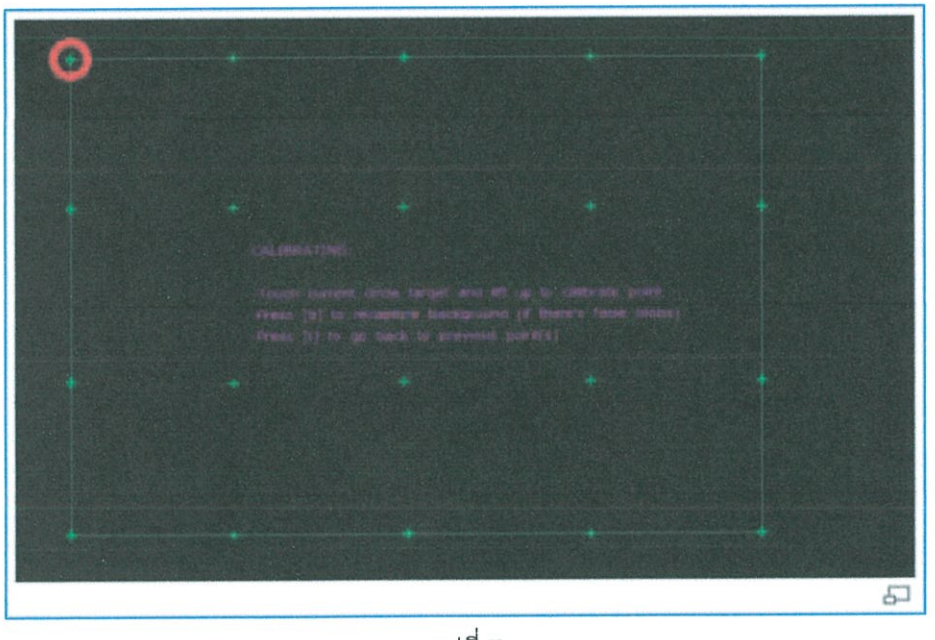
รูปที่ 5

7. สามารถปรับขอบเขตของการ Calibration ได้โดยการกดอักษร “W” บนแป้นคีย์บอร์ด เพื่อปรับขึ้นด้านบน, อักษร “A” บนแป้นคีย์บอร์ด เพื่อปรับไปทางซ้าย, อักษร “D” บนแป้นคีย์บอร์ด เพื่อปรับไปทางขวา และอักษร “S” บนแป้นคีย์บอร์ด เพื่อปรับลงด้านล่าง ตามรูปที่ 6



รูปที่ 6

8. เริ่มต้น Calibration โดยการกดอักษร “C” บนแป้นคีย์บอร์ด อีกครั้งนึง และนำนิ้วหรือวัตถุไปสัมผัสตามจุดสีแดงที่ปรากฏบนหน้าจอ หากระหว่างกระบวนการเกิดข้อผิดพลาดสามารถกดอักษณ “R” บนแป้นคีย์บอร์ด เพื่อย้อนกลับไปที่จุดเดิมอีกครั้ง ตามรูปที่ 7



รูปที่ 7

9. กดอักษร “X” บนแป้นคีย์บอร์ด เพื่อออกจากโหมด Calibration