การอินเตอร์เฟสกับเกมโดยวิธีการประมวลผลกับวีดีโอ

Game Interface Using Video Processing

นายภากร น้อยสอน รหัส 49371231

นางสาวสุภาวดี พาพิมพ์ รหัส 49371392

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 11 ส.ค. 2555
เลขทะเบียน 157347198
เลขเรียกหนังสือ ว.ร.
มหาวิทยาลัยนเรศวร ป 411ก 2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553

ใบรับรองโครงงานวิศวกรรม

ชื่อหัวข้อโครงงาน การอินเตอร์เฟสกับเกมโดยวิธีการประมวลผลกับวีดีโอ

ผู้ดำเนินโครงงาน นายภากร น้อยสอน รหัส 49371231

นางสาวสุภาวดี พาพิมพ์ รหัส 49371392

อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.ไพศาล มุณีสว่าง

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ที่ปรึกษาโครงงาน

(รองศาสตราจารย์ ดร.ไพศาล มุณีสว่าง)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

.....กรรมการ

(ดร.พรพิศุทธิ์ วรจิรันตน์)

หัวข้อโครงงาน	การอินเตอร์เฟสกับเกมโดยวิธีการประมวลผลกับวีดีโอ
ผู้ดำเนินโครงงาน	นายภากร น้อยสอน รหัส 49371231
	นางสาวสุภาวดี พาพิมพ์ รหัส 49371392
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.ไพศาล มุณีสว่าง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

.....

บทคัดย่อ

ปัจจุบันเกมคอมพิวเตอร์ได้รับความนิยมเป็นอย่างมาก ไม่ว่าจะเป็นเกมบนเครือข่ายอินเตอร์เน็ต หรือเกมแบบออฟไลน์ แต่ส่วนใหญ่การเล่นเกมแบบเดิม ผู้เล่นจะต้องใช้เมาส์แป้นพิมพ์ หรือจอยสติกในการเล่น ทำให้ผู้เล่นมีปฏิสัมพันธ์กับเกมไม่มากนัก อีกทั้งเทคโนโลยีด้านการประมวลผลภาพได้รับความนิยมในการนำมาใช้พัฒนาเกมเพิ่มสูงขึ้น ดังนั้นผู้จัดทำจึงได้ศึกษาหาวิธีการที่ใช้ในการควบคุมเกมแบบใหม่ โดยได้นำหลักการเรื่องการประมวลผลภาพมาใช้วิเคราะห์ภาพที่ได้จากกล้องวีดีโอ และมุ่งเน้นไปที่การวิเคราะห์หาส่วนมือของผู้เล่น เพื่อนำมาใช้อินเตอร์เฟสกับเกม ให้สามารถใช้มือบังคับตัวละครในเกมได้ เพื่อให้เกิดการปฏิสัมพันธ์กับเกมและเพิ่มความสนุกสนานในการเล่นเกมมากขึ้น

Project title	Game Interface Using Video Processing
Name	Mr. Parkorn Noison ID. 49371231
	Miss. Suphawadi Phapim ID. 49371392
Project advisor	Assoc. Prof. Dr. Paisarn Muneesawang
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2010

.....

ABSTRACT

At present computer gaming is very popular. Both online gaming and on LAN gaming. They have common controlling methods such as using mouse, keyboard, joystick, etc. These methods may not attractive for players. So, a new method of controlling using the theory of image processing, which analyzes the image that we get from video camera and focus on the hand of player. The player controls the game by moving a hand, which is more entertaining and enjoyable.

กิตติกรรมประกาศ

ปริญญานิพนธ์การอินเตอร์เฟสกับเกมโดยวิธีประมวลผลกับวีดีโอนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากความกรุณาจาก รองศาสตราจารย์ ดร.ไพศาล มุณีสว่าง อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานนี้ และให้ความกรุณาเสียสละเวลาและความเอื้อเฟื้อ แก่ผู้ดำเนินโครงงาน อีกทั้งยังคอยช่วยให้คำแนะนำแก้ไขปัญหาในการปฏิบัติงาน คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินงานเหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณคุณพ่อ คุณแม่ ผู้มอบความรักความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอขอบคุณทุกๆคนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นายภาคร น้อยสอน
นางสาวสุภาวดี พาพิมพ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบข่ายโครงงาน 2

1.4 ขั้นตอนการดำเนินงาน 2

1.5 ผลที่คาดหวัง 3

1.6 งบประมาณที่ใช้ 3

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 4

2.1 คอมพิวเตอร์วิทัศน์ (Computer Vision) 4

2.2 การประมวลผลภาพด้วยวีดีโอ (Video processing) 4

2.3 การประมวลผลภาพด้วยคอมพิวเตอร์ (Image Processing) 5

2.4 การติดตามวัตถุ (Object tracking) 5

2.4.1 แบบโมเดล (Model-based) 5

2.4.2 แบบพื้นที่ (Region-based) 6

2.4.3 แบบขอบวัตถุ (Contour-based) 6

2.4.4 แบบลักษณะเฉพาะ (Feature-based) 6

2.5 ระบบสี RGB 6

2.6 การแยกวัตถุออกจากพื้นหลัง (Thresholds) 7

2.7 การแยกจุดเด่นของภาพ (ColorIsolator) 9

2.8 โปรแกรม Adobe Flash CS5 10

สารบัญ(ต่อ)

หน้า

2.8.1 ซิมโบลและอินสแตนซ์ 10
2.8.2 ประเภทของไฟล์ Flash 10
2.9 ภาษา ActionScript 3.0 11
2.10 โปรแกรม ColorTracking 11
2.10.1 บล็อกไดอะแกรมของโปรแกรม ColorTracking 11
2.10.2 ส่วนขยายของโปรแกรม ColorTracking 12
2.10.2.1 การติดต่อกับกล้องเว็บแคม 12
2.10.2.2 คลาสไดอะแกรมสำหรับการติดต่อกับกล้องเว็บแคม 13
2.10.2.3 คลาสไดอะแกรมสำหรับการแสดงผลวีดีโอเป็นแบบบิตแมป 15
2.10.2.4 ปรับปรุงคุณภาพของสี (Color Enhancement) 17
2.10.2.5 สรุปขั้นตอนการปรับปรุงคุณภาพของสี 23
2.10.2.6 การตรวจจับสี 24
2.10.2.7 กำหนดขอบเขตพื้นที่ของสีโดยใช้ getColorBoundRect 26
2.11 การแปลงรูปภาพระดับพิกเซล (BitmapData Class) 27
2.12 การจัดการเกี่ยวกับสี (ColorMatrix) 27
2.12.1 การแปลงสีให้เป็นระบบเมตริกซ์ 28
2.13 แปลงค่าเมตริกซ์ (ColorMatrixFilter) 29
2.14 ไลบรารี่ที่ใช้สำหรับการทำ UI (Library bit101) 29
2.15 การทำให้ออบเจ็กต์มีการเปลี่ยนแปลง (Tweening) 30
2.16 การทำให้กรอบเคลื่อนที่ตามเป้าหมาย (Library TweenLite) 30

บทที่ 3 รายละเอียดการดำเนินงาน 32
3.1 ขั้นตอนการดำเนินงาน 32
3.2 ออกแบบเกม 32
3.2.1 ออกแบบโมเดลที่ใช้ในเกม 33
3.2.2 การออกแบบเงื่อนไขต่างๆของเกม 39
3.2.3 เขียนโปรแกรม 39
3.2.4 วิธีการเล่นเกม 46

สารบัญ(ต่อ)

หน้า

3.3 ปรับปรุงโปรแกรม ColorTracking ให้เหมาะสมกับเกม 46
3.3.1 ปรับปรุงการแสดงผลบนหน้าจอ 46
3.3.1.1 ปรับขนาดการแสดงผลของหน้าจอ 46
3.3.1.2 ปรับขนาดและตำแหน่งการแสดงผลของกล้องเว็บแคม 47
3.3.1.3 นำปุ่มและส่วนที่ใช้ปรับภาพทั้งหมดออก 48
3.3.2 เขียนฟังก์ชั่นให้โมเดลที่เตรียมไว้ติดตามการเคลื่อนที่ของมือ 51
3.3.2.1 เขียนฟังก์ชั่นของโมเดลมือ 52
3.3.2.2 เขียนฟังก์ชั่นของโมเดลยานอวกาศ 53
3.3.3 คลาสไดอะแกรมของคลาสหลัก 54
3.4 Publish โปรแกรม 55
3.5 ทฤษฎีและหลักการต่างๆที่นำมาใช้ในขั้นตอนการตรวจจับและติดตามสี 55

บทที่ 4 ผลการทดสอบโปรแกรม 56
4.1 ผลการทดสอบความแม่นยำในการประมวลผล 56
4.1.1 รูปภาพแสดงการประมวลผลภาพจากการทดลอง 57
4.2 ผลการทดลองขั้นตอนการทำงานของเกม 59

บทที่ 5 สรุปผลและข้อเสนอแนะ 63
5.2 สรุปผล..... 63
5.2 ปัญหาและแนวทางแก้ไข 63
5.3 ข้อเสนอแนะ/แนวทางการพัฒนาต่อ 64

เอกสารอ้างอิง 65
ประวัติผู้เขียนโครงงาน 66

สารบัญตาราง

ตารางที่ หน้า

1.1 ระยะเวลาและกิจกรรมในการดำเนินโครงงาน 3
3.1 อธิบายส่วนต่างๆในคลาสไดอะแกรม Ship 41
3.2 อธิบายส่วนต่างๆในคลาสไดอะแกรม Stinger 42
3.3 อธิบายส่วนต่างๆในคลาสไดอะแกรม LaserBlue 44
3.4 อธิบายส่วนต่างๆในคลาสไดอะแกรม StingerBullet 45
3.5 ฟังก์ชั่นที่ต้องเขียนเพิ่มเข้าไปในคลาส ColorTracking 45
3.6 สรุปทฤษฎีและหลักการต่างๆที่นำมาใช้ในขั้นตอนการตรวจจับและติดตามสี 55

สารบัญรูป

รูปที่ หน้า

2.1 RGB Color Model 7

2.2 แสดง histogram ของ image 7

2.3 ผลของการใช้ Global Thresholding 8

2.4 ตัวอย่าง Automatic Thresholding 9

2.5 ตัวอย่าง ตัวอย่างภาพที่ผ่านการทำ ColorIsolation 9

2.6 บล็อกไดอะแกรมของโปรแกรม ColorTracking 11

2.7 การขออนุญาตเชื่อมต่อกับกล้องเว็บแคม 12

2.8 คลาสไดอะแกรมสำหรับการติดต่อกับกล้องเว็บแคม 13

2.9 คลาสไดอะแกรมสำหรับการแสดงผลวีดีโอแบบบิตแมป 15

2.10 แสดงการสร้างสไลเดอร์ในการปรับภาพในฟังก์ชั่น setupControl 18

2.11 UI สำหรับปรับปรุงภาพ 20

2.12 ภาพวีดีโอจากการเซตค่าสีของโปรแกรม 21

2.13 สรุปขั้นตอนการปรับปรุงคุณภาพของสี 24

2.14 ผลที่ได้หลังจากการใช้เมธอด threshold 25

2.15 หลังจากใช้เมธอด getColorBoundsRect ติดตามสีที่ต้องการ 27

2.16 ค่าดีฟอลต์ของ ColorMatrix 28

2.17 การคำนวนด้วย ColorMatrix 29

2.18 ส่วนประกอบต่างๆของ Library bit101 30

3.1 การสร้างโมเดลรูปมือและการตั้งชื่อ 33

3.2 โมเดลรูปมือหลังจากใช้เครื่องมือในการวาด 34

3.3 การสร้างซิมโบลยานอวกาศ 34

3.4 สร้าง hit area 35

3.5 การทำให้ยานอวกาศระเบิด 35

3.6 สร้างเกราะป้องกัน 36

3.7 การกำหนดชื่อ label ลงบนเฟรม 36

3.8 รูปร่างโมเดลยานอวกาศฝ่ายตรงข้าม 37

3.9 ฉากหลักของเกม 37

3.10 ภาพโดยรวมของเกม 38

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.11 อาวุธของยานอวกาศตัวหลักและฝ่ายตรงข้าม 38

3.12 ตารางแสดงคะแนน 38

3.13 ฉากเริ่มต้น ฉากจบและเมนูต่างๆ 38

3.14 Flow Chart ของเกม 39

3.15 คลาสไดอะแกรมของยานอวกาศฝ่ายผู้เล่น 40

3.16 คลาสไดอะแกรมของยานอวกาศฝ่ายตรงข้าม 42

3.17 คลาสไดอะแกรมของการยิงเลเซอร์ฝ่ายผู้เล่น 43

3.18 คลาสไดอะแกรมของการยิงระเบิดของฝ่ายตรงข้าม 44

3.19 การกำหนดขนาดจอแสดงผล 47

3.20 การแสดงผลของโปรแกรม ColorTracking 47

3.21 block1, block2 47

3.22 การปรับให้เหลือเพียง block1 48

3.23 จอภาพก่อนและหลังการนำส่วนที่ไม่เกี่ยวข้องออก 48

3.24 รูปภาพวีดีโอก่อนและหลังจากมีการปรับปรุงภาพ 50

3.25 การตรวจจับสีของมือ 51

3.26 การกำหนดกรอบสี่เหลี่ยมในการติดตามการเคลื่อนที่ของสี 52

3.27 แสดงผลการใช้กรอบสี่เหลี่ยมติดตามการเคลื่อนที่ของมือ 52

3.28 การใช้โมเดล hand ในการติดตามการเคลื่อนที่ของมือ 53

3.29 การใช้โมเดล ourShip ในการติดตามการเคลื่อนที่ของมือ 54

3.30 Block Diagram ของการปรับปรุงโปรแกรมเพื่อนำมาใช้กับเกม 54

3.31 ไฟล์ .swf ที่ได้จากการพับลิชโปรแกรม 55

4.1 การทดสอบการเล่นเกมในห้องที่มีความสว่างมากหรือกล้องอยู่ในตำแหน่งย้อนแสง 57

4.2 การทดลองเล่นเกมในห้องที่มีความสว่างน้อยเกินไป 58

4.3 การทดสอบเล่นเกมในสถานที่ที่ด้านหลังมีวัตถุอื่นๆปะปน 58

4.4 การทดสอบเล่นเกมโดยติดตั้งกล้องเว็บแคมไว้ในระดับที่มองเห็นใบหน้า 59

4.5 ผลลัพธ์เมื่อเปิดโปรแกรม 60

4.6 ผลลัพธ์เมื่อเปิดกล้องเว็บแคม 60

4.7 Symbol hand เคลื่อนที่ไปตามมือ 61

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.8 การใช้มือบังคับตัวละครในเกม 61

4.9 Game Over 62

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เกมคอมพิวเตอร์ได้รับความนิยมมาตั้งแต่อดีตจนถึงปัจจุบัน และมีการพัฒนาขึ้นเป็นลำดับ ค่ายเกมต่างๆ ล้วนแล้วแต่แข่งขันกันผลิตเกมใหม่ๆขึ้นมา เพื่อสร้างความแปลกใหม่ให้กับผู้เล่นเกม การที่จะสร้างเกมขึ้นมาใหม่จึงจำเป็นจะต้องคิดค้นหาวิธีการที่จะทำให้เกมมีความน่าสนใจมากขึ้น และควรจะมีประโยชน์ต่อผู้เล่นเกมมากขึ้น เพื่อทำให้เกมเป็นที่ยอมรับและเป็นที่นิยมมากกว่าเดิม ซึ่งในปัจจุบันในต่างประเทศหลักการด้านการประมวลผลภาพกำลังเป็นที่นิยมสำหรับการพัฒนาเกม ซึ่งมีการนำเอาหลักการต่างๆมาใช้แตกต่างกันไป มีการศึกษาค้นคว้าวิธีการนำมาใช้มากมาย เกมที่ได้จากการประมวลผลจะเป็นเกมที่ใช้กล้องวีดีโอหรือกล้องเว็บแคมในการควบคุมการเล่น ซึ่งก่อให้เกิดการโต้ตอบระหว่างมนุษย์กับคอมพิวเตอร์มากขึ้น

เกมเป็นสื่อบันเทิงที่ก่อให้เกิดความสนุกสนาน รื่นเริงใจกับผู้เล่น และบางทีอาจเป็นการพัฒนาสมองได้สำหรับในบางเกม การเล่นเกมยังช่วยคลายความเครียดจากการเรียน การทำงานต่างๆ และลดความวิตกกังวลลงได้ แต่ในบางครั้งเกมก็ก่อให้เกิดปัญหาต่างๆถ้าหากผู้เล่นไม่พิจารณาให้ดีว่าสิ่งไหนคือเกม สิ่งไหนคือชีวิตจริงซึ่งส่วนใหญ่จะเกิดขึ้นกับผู้เล่นในวัยเด็ก บางคนอาจจะติดเกมมากเกินไปจนเสียการงาน การเรียน บางครั้งนั่งเล่นนานเกินไปก็อาจทำให้เสียสุขภาพได้ ดังนั้นการออกแบบเกมให้เหมาะสมจึงเป็นสิ่งจำเป็นมาก โดยส่วนมากแล้วเกมทั่วไปจะไม่ค่อยได้เคลื่อนไหวส่วนต่างๆของร่างกายนัก ถ้าเล่นเกมแล้วได้เคลื่อนไหว ออกกำลังกายไปด้วยจะทำให้การเล่นเกมมีประโยชน์มากกว่าเดิม นอกจากได้ความสนุกสนานแล้วยังช่วยให้สุขภาพแข็งแรงได้อีกด้วย เกมจะมีคุณลักษณะและจำนวนผู้เล่นที่แตกต่างกันออกไปตามจุดประสงค์ของเกมนั้น ส่วนใหญ่จะเป็นการแข่งขันระหว่างผู้เล่นสองคนหรือสองฝ่าย ซึ่งแบ่งเป็นหลายประเภท เช่น เกมผจญภัย, เกมกีฬา, เกมกลยุทธ์, เกมการ์ด, เกมปริศนา, เกมสงคราม เป็นต้น

ซึ่งเกมในปัจจุบันส่วนใหญ่จะใช้ เมาส์ แป้นพิมพ์ หรือจอยสติก ในการควบคุมเกม ซึ่งเป็นเกมที่มีรูปแบบเดิมๆ ผู้เล่นไม่ค่อยมีปฏิสัมพันธ์กับเกมมากนัก เนื่องจากจะต้องนั่งอยู่กับที่ตลอดเวลาและอาจทำให้เสียสุขภาพได้ ผู้ทำโครงงานได้เล็งเห็นว่าน่าจะเพิ่มความน่าสนใจให้กับเกมคือเปลี่ยนรูปแบบในการเล่นเกมแบบเดิม เป็นการใช้แค่มือเปล่าในการควบคุม เพื่อเพิ่มความสนุกสนานและการมีปฏิสัมพันธ์กับเกมให้กับผู้เล่น อีกทั้งเทคโนโลยีด้านการประมวลผลภาพกำลังเป็นที่นิยมและเป็นที่สนใจ ในการนำมาใช้พัฒนาเกมในรูปแบบใหม่ๆ ในปัจจุบัน และยังเป็นที่นิยม

ในการนำมาใช้พัฒนาในด้านอื่นๆอีกเช่น ใช้ในการตรวจจับป้ายทะเบียนรถเพื่อช่วยส่งเสริมในด้านการรักษาความปลอดภัย, ใช้ในการวิเคราะห์ภาพใบหน้าบุคคลเพื่อระบุตัวบุคคล เป็นต้น จึงเล็งเห็นว่าควรจะศึกษาหลักการการประมวลผลภาพเพื่อนำมาใช้ในการพัฒนาเกมต่อไป

โครงงานนี้จะศึกษาเกี่ยวกับอัลกอริทึมที่ใช้ติดตามการเคลื่อนที่ของมือ และหาวิธีการที่จะนำมาอินเตอร์เฟสกับเกมให้สามารถควบคุมตัวละครได้ เน้นการจับภาพมือของผู้เล่นผ่านกล้องเว็บแคม จากนั้นนำมาอินเตอร์เฟสกับเกม โปรแกรมที่ใช้ในการอินเตอร์เฟสคือโปรแกรม Flash cs5 ซึ่งเป็นโปรแกรมสำหรับเขียนแฟลชเกม มีฟังก์ชั่นรับรองการประมวลผลภาพทำการเชื่อมต่อกันและเรียกใช้งานด้วยภาษา Action script 3.0 เกมที่สร้างเป็นเกมที่ไม่ซับซ้อน สามารถเล่นได้ทุกเพศทุกวัย

1.2 วัตถุประสงค์ของโครงงาน

วัตถุประสงค์หลักของโครงงานนี้คือ การนำหลักการการประมวลผลภาพมาใช้ในการวิเคราะห์หาส่วนมือในภาพที่ได้รับจากกล้องวีดีโอ แล้วนำมาอินเตอร์เฟสกับส่วนของเกมให้สามารถควบคุมเกมได้

1.3 ขอบข่ายโครงงาน

1.3.1 การค้นหาภาพมือจากกล้องวีดีโอในระยะทางโดยประมาณ 1-2 เมตร

1.3.2 มีการพัฒนาภาพมือในระหว่างการตรวจจับ เพื่อให้วิเคราะห์ได้ชัดเจนขึ้น

1.3.3 กล้องที่ใช้ในการทดสอบมีความคมชัด 1.3 M ขึ้นไป

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาโปรแกรม Flash CS5 และ Action Script 3.0
2. ศึกษาการเขียน Action Script เพื่อใช้ในกระบวนการประมวลผลภาพ
3. ศึกษาโปรแกรม ColorTracking
4. ปรับปรุงโปรแกรม ColorTracking ให้สามารถนำมาใช้ในการอินเตอร์เฟสกับเกมได้
5. ตรวจจับหาสีและติดตามการเคลื่อนที่ของมือ
6. การอินเตอร์เฟสและการออกแบบ
7. ทดสอบเกม

ตารางที่ 1.1 ระยะเวลาและกิจกรรมในการดำเนินโครงงาน

กิจกรรม	ปี 2553							ปี 2554			
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
ศึกษาโปรแกรม Flash CS5 และ Action Script 3.0	←	→									
ศึกษาการเขียน Action Script เพื่อใช้ในกระบวนการประมวลผลภาพ		←	→								
ศึกษาโปรแกรม ColorTracking					←	→					
ปรับปรุงโปรแกรม ColorTracking							↔				
ตรวจจับหาสีและติดตามการเคลื่อนที่ของมือ								←	→		
การอินเตอร์เฟสและการออกแบบ										←	→
ทดสอบเกม											↔

1.5 ผลที่คาดหวัง

1.5.1 ได้ขั้นตอนการวิเคราะห์ภาพเคลื่อนไหวและการวิเคราะห์ภาพในส่วนที่ต้องการโดยวิธีการแยกคุณลักษณะของสีในภาพวีดีโอ

1.5.2 ได้ขั้นตอนและวิธีการนำการประมวลผลภาพมาอินเตอร์เฟสกับเกม

1.5.3 ได้เกมในรูปแบบใหม่ๆเพิ่มขึ้น

1.6 งบประมาณที่ใช้

1.6.1 ค่าวัสดุและอุปกรณ์	1,000	บาท
1.6.2 ค่าทำเอกสารเป็นเงิน	1,000	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาทถ้วน

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

การพัฒนาโปรแกรมการประมวลผลภาพเพื่อนำมาอินเตอร์เฟสกับเกมให้สามารถควบคุมตัวละครในเกมได้นั้น จำเป็นต้องใช้ความรู้ในหลายๆส่วนประกอบกัน ไม่ว่าจะเป็นด้านการประมวลผลภาพ, โครงสร้างของโปรแกรม ColorTracking ที่นำมาใช้ในการศึกษาการตรวจจับและติดตามการเคลื่อนที่, การเขียนโปรแกรมด้วยภาษา ActionScript 3.0 รวมถึงความรู้ในด้านการสร้างเกมด้วยโปรแกรม Adobe Flash CS5 และ ไลบรารีต่างๆที่นำมาใช้ในการทำโครงงาน

การพัฒนาโปรแกรมการประมวลผลภาพจากกล้องวีดีโอ เพื่อนำมาอินเตอร์เฟสกับเกมให้สามารถควบคุมตัวละครในเกมได้นั้นต้องมีความรู้และทราบหลักการที่เกี่ยวข้องเพื่อนำมาใช้ประกอบในการสร้างโปรแกรม ซึ่งหลักการและทฤษฎีที่สำคัญทั้งหมดได้อธิบายไว้ดังต่อไปนี้

2.1 คอมพิวเตอร์วิทัศน์ (Computer Vision)

คอมพิวเตอร์วิทัศน์(Computer Vision)[2] หมายถึงวิทยาศาสตร์และเทคโนโลยีของคอมพิวเตอร์ เป้าหมายของคอมพิวเตอร์วิทัศน์โดยทั่วไป คือ การทำให้คอมพิวเตอร์สามารถเข้าใจรูปที่รับเข้ามา โดยการแปลงจากภาพ (image) เป็น แบบจำลอง (model) ที่คอมพิวเตอร์สามารถรับรู้คุณสมบัติที่ต้องการได้ เช่น คุณสมบัติทางรูปร่าง เพื่อการจดจำ เพื่อการหยิบจับ เป็นต้น

กระบวนการจะเกี่ยวเนื่องตั้งแต่ การรับภาพ (ซึ่งไม่ได้จำกัดอยู่เพียงแต่รูปที่รับด้วยกล้อง แต่อาจหมายถึงภาพถ่ายความร้อน และ ภาพจากอุปกรณ์วัดระยะ ฯลฯ) การประมวลผลภาพ การสร้างแบบจำลองจากภาพ และการรับรู้เข้าใจของคอมพิวเตอร์จากแบบจำลอง

2.2 การประมวลผลภาพด้วยวีดีโอ (Video processing)

Video processing เป็นหลักการประมวลผลจากสัญญาณวีดีโอ[3] หลักการมีลักษณะเช่นเดียวกับ Digital image processing และ Digital image analysis

Digital Image processing คือ การแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอล (Digital format) ซึ่งสามารถที่จะนำเอาข้อมูลนี้จัดผ่านกระบวนการต่างๆ ด้วยดิจิตอลคอมพิวเตอร์ได้ ในระบบของดิจิตอล อินพุตและเอาต์พุตของระบบจะอยู่ในรูปแบบดิจิตอลเท่านั้น

Digital image analysis จะเกี่ยวกับวิธีการอธิบาย และการจดจำข้อมูลภาพดิจิตอล ซึ่งอินพุตของระบบจะเป็นข้อมูลภาพดิจิตอล และเอาต์พุตจะเป็นเครื่องหมายที่ใช้แทนข้อมูลภาพดิจิตอลเหล่านั้น ในการวิเคราะห์ภาพมีอยู่หลายวิธีด้วยกันที่ได้นำมาจากการทำงานของตามนุษย์(human vision)นั่นก็คืองานทางด้าน Computer Vision เป็นลักษณะเดียวกับ Digital image analysis นั่นเอง

การมองเห็นของมนุษย์นับว่าเป็นกระบวนการที่ซับซ้อนซึ่งลักษณะเทคนิคโดยทั่ว ๆ ไปในกระบวนการ Digital image analysis และ Computer Vision จะค่อนข้างซับซ้อนเช่นกันในระบบของดิจิตอล อินพุตและเอาต์พุตของระบบจะอยู่ในรูปแบบดิจิตอลเท่านั้น

2.3 การประมวลผลภาพด้วยคอมพิวเตอร์ (Image Processing)

Image Processing คือ การประมวลผลภาพด้วยคอมพิวเตอร์[4] เพื่อให้คอมพิวเตอร์ทราบว่าภาพนั้นคือภาพอะไร หรือมีสิ่งที่สนใจอยู่ในภาพหรือไม่ โดยที่ไม่ต้องใช้สายตาของคนมาช่วยตัดสิน การคิดคำนวณนั้นมีหลายวิธีซึ่งแต่ละวิธีก็มีประโยชน์แตกต่างกันไป ไม่ว่าจะเป็นการนำเอาสีแต่ละจุด (Pixel) มาคิด (Color), การคิดคำนวณเป็นบริเวณหลายๆจุดรวมๆกัน (Area) เช่นการดูลวดลาย (Pattern, Texture), การวิเคราะห์หารูปร่าง (Shape) และการวิเคราะห์แบบอื่นๆ เพื่อหาค่าที่สามารถระบุได้ว่าภาพนั้นมีลักษณะอย่างไร ซึ่งจะนำไปสู่การตัดสินได้ว่าสิ่งนั้นเป็นสิ่งที่กำลังค้นหาหรือสนใจอยู่หรือไม่

ในโครงงานชิ้นนี้ได้นำความรู้เรื่อง Image Processing มาใช้ประโยชน์ในด้านการประมวลผลภาพจากกล้องวีดีโอ ซึ่งจะนำมาวิเคราะห์ตำแหน่งของแต่ละจุดสี (Pixel) ในรูปภาพแต่ละรูปนั้นจะมีการแบ่งเป็นแถว (row) และคอลัมน์ (column) และในตำแหน่ง x, y ใด ๆ นั้น จะเก็บค่าต่างๆเอาไว้ ได้แก่ ค่าความเข้มของแสงและเวลา ทำให้ต้องใช้กระบวนการจัดการเกี่ยวกับรูปภาพเข้ามาเกี่ยวข้อง

2.4 การติดตามวัตถุ (Object tracking)

การติดตามวัตถุ (Object tracking)[5] เป็นเทคนิคหนึ่งที่มีความสำคัญมากในงานด้านความปลอดภัยโดยอาศัยศาสตร์ด้าน Image processing ภาพหนึ่งภาพมีข้อมูลอยู่มากมาย การที่จะดึงข้อมูลที่ต้องการออกมา (Image segmentation and extraction) โดยแยกจากข้อมูลด้านอื่นๆ รวมไปถึงสิ่งรบกวนต่างๆ (Noises) ที่ไม่ต้องการ ต้องอาศัย algorithm และขั้นตอนที่เหมาะสม

การติดตามวัตถุที่เคลื่อนที่ (Object Tracking) เป็นสิ่งจำเป็นมากเพราะทำให้ทราบถึงทิศทางการเคลื่อนที่ และการกระทำ ซึ่งจะนำไปสู่การรู้จำวัตถุนั้นๆ นักวิจัยหลายท่านได้นำเสนอการการติดตามวัตถุที่เคลื่อนที่ ซึ่งโดยทั่วไปแล้วสามารถจำแนกออกได้เป็น 4 ประเภทคือ

2.4.1 แบบโมเดล (Model-based)

การติดตามวัตถุที่เคลื่อนที่แบบโมเดล เป็นความพยายามที่จะติดตามวัตถุที่มีรูปแบบเข้ากับโมเดลที่เตรียมไว้ ซึ่งโมเดลเหล่านี้มักจะเตรียมไว้ก่อน และต้องรู้ล่วงหน้า ดังนั้นจึงมีข้อด้อยที่ไม่สามารถทำงานกับวัตถุที่ไม่ทราบล่วงหน้าได้

2.4.2 แบบพื้นที่ (Region-based)

การติดตามวัตถุที่เคลื่อนที่แบบพื้นที่นั้น เป็นการดึงข้อมูลสำคัญเช่น สี ลวดลาย จากนั้นทำการติดตามพื้นที่เหล่านั้นโดยหาข้อมูลของสิ่งที่ดึงออกมาใช้

2.4.3 แบบขอบวัตถุ (contour-based)

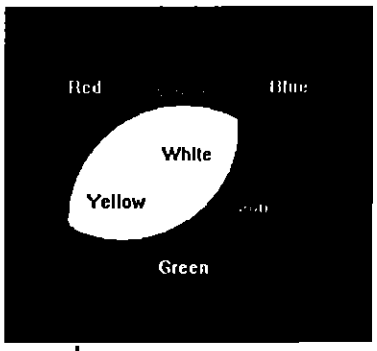
การติดตามวัตถุที่เคลื่อนที่แบบขอบวัตถุนั้น จะขึ้นกับขอบของวัตถุที่เคลื่อนที่ ไม่ได้ขึ้นกับตัววัตถุทั้งชิ้น ขอบวัตถุจะถูกสกัดออกมาถูกปรับแต่งในแต่ละเฟรมที่เกิดขึ้น ดังนั้นการติดตามวัตถุที่เคลื่อนที่จะได้ผลดีหรือไม่ขึ้นกับขอบวัตถุเริ่มแรกที่ถูกสกัดออกมาได้ ซึ่งวิธีนี้ยากต่อการนำมาใช้ในระบบตรวจจับเพื่อรักษาความปลอดภัย

2.4.4 แบบลักษณะเฉพาะ (Feature-based)

สำหรับการติดตามวัตถุที่เคลื่อนที่แบบลักษณะเฉพาะนั้น มีจุดประสงค์ที่จะค้นหาและติดตามการติดตามวัตถุที่เคลื่อนที่โดยใช้คุณลักษณะเฉพาะบางอย่างเช่น เส้นรอบวง พื้นที่ของวัตถุ ขอบ มุม เป็นต้น ซึ่งวิธีการนี้จะให้ผลไม่ดีหากวัตถุถูกบดบังบางส่วน

2.5 ระบบสี (RGB)

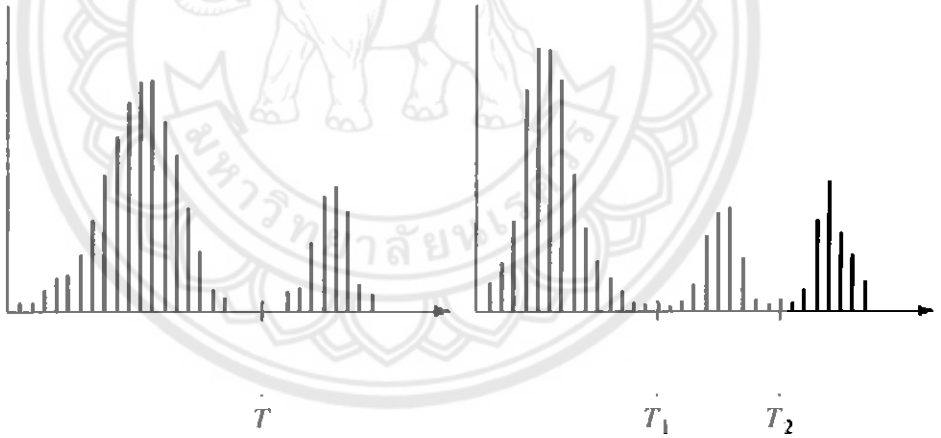
ระบบสี (RGB)[6] เป็นระบบสีของแสง ซึ่งเกิดจากการหักเหของแสงผ่านแท่งแก้วปริซึม จะเกิดแถบสีที่เรียกว่า สีรุ้ง (Spectrum) ซึ่งแยกสีตามที่สายตามองเห็นได้ 7 สี คือ แดง แสด เหลือง เขียว น้ำเงิน คราม ม่วง ซึ่งเป็นพลังงานอยู่ในรูปของรังสี ที่มีช่วงคลื่นที่สายตา สามารถมองเห็นได้ แสงสีม่วงมีความถี่คลื่นสูงที่สุดคลื่นแสงที่มีความถี่สูงกว่าแสงสีม่วง เรียกว่า อัลตราไวโอเลต (Ultra Violet) และคลื่นแสงสีแดง มีความถี่คลื่นต่ำที่สุด คลื่นแสง ที่ต่ำกว่าแสงสีแดงเรียกว่า อินฟราเรด (Infrared) คลื่นแสงที่มีความถี่สูงกว่าสีม่วง และต่ำกว่าสีแดงนั้น สายตาของมนุษย์ไม่สามารถรับได้ และเมื่อศึกษาดูแล้วแสงสีทั้งหมดเกิดจาก แสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue) และสีเขียว (Green) ทั้งสามสีถือเป็นแม่สีของแสง เมื่อนำมาฉายรวมกันจะทำให้เกิดสีใหม่อีก 3 สี คือ สีแดงมาเจนต้า สีฟ้าไซแอน และสีเหลือง และถ้าฉายแสงสีทั้งหมดรวมกันจะได้แสงสีขาว จากคุณสมบัติของแสงนี้เรา ได้นำมาใช้ประโยชน์ทั่วไป ในการฉายภาพยนตร์ การบันทึกภาพวิดีโอ ภาพโทรทัศน์ การสร้างภาพเพื่อการนำเสนอทางจอคอมพิวเตอร์ และการจัดแสงสีในการแสดง เป็นต้น



รูปที่ 2.1 RGB Color Model

2.6 การแยกวัตถุออกจากพื้นหลัง (Thresholds)

การกำหนด Threshold เป็นค่าความเข้มให้มีค่าที่สามารถแยกความแตกต่างของวัตถุและพื้นหลังได้ตัวอย่างอย่างเช่น ภาพของตัวอักษรที่มีความความเข้มของตัวอักษรเป็น 0 (สีดำ) และมีความเข้มของพื้นหลังเป็น 255 (สีขาว) ดังนั้นค่า Threshold จึงควรจะมีค่าเท่ากับ 128 เพื่อที่จะให้สามารถแยกวัตถุออกจากพื้นหลังได้ โดยปกติแล้วการเลือกค่า Threshold จะขึ้นอยู่กับ Histogram ของภาพ



รูปที่ 2.2 แสดง histogram ของ image

จากรูป 2.2 (a) image f(x, y) ประกอบด้วยวัตถุ (object) สีอ่อน และพื้นหลัง (background) สีเข้ม จึงเห็นกลุ่มของการกระจาย gray level เป็นสองกลุ่มอย่างชัดเจน เมื่อต้องการแยกวัตถุออกจากพื้นหลังอาจจะทำได้โดยเลือกค่า threshold (T) เช่น

f(x, y) > T จะกำหนดให้เป็นส่วนของวัตถุ

f(x, y) >= T จะกำหนดให้เป็นส่วนของพื้นหลัง

จากรูป 2.2 (b) image f(x, y) ประกอบด้วยการกระจาย gray level เป็นสามกลุ่ม เช่นมีวัตถุสองชนิดสีอ่อนอยู่บนพื้นหลังสีเข้ม เมื่อต้องการแยกวัตถุออกจากพื้นหลังอาจจะทำได้โดยใช้ multilevel thresholding เช่น

T1 < f(x, y) <= T2 จะกำหนดให้เป็นส่วนของวัตถุที่หนึ่ง

f(x, y) > T2 จะกำหนดให้เป็นส่วนของวัตถุที่สอง

f(x, y) <= T1 จะกำหนดให้เป็นส่วนของพื้นหลัง

เขียนสมการทั่วไปของผลลัพธ์ที่ได้จากกระบวนการ thresholding ได้เป็น

$$g(x,y) = \begin{cases} 1 & f(x,y) > T \\ 0 & f(x,y) < T \end{cases}$$

Basic Global Thresholding แบ่งวัตถุออกจากพื้น หลังโดยเลือกใช้ค่า threshold 1 ค่า

รูปที่ 2.3 ผลของการใช้ Global Thresholding

Basic automatic thresholding

การเลือกค่า T อาจจะใช้วิธีแบบอัตโนมัติดังเช่น algorithmต่อไปนี้

1 เลือกค่าเริ่มต้นให้กับ T อาจจะเป็น

a. ค่าเฉลี่ยของ gray level ใน image หรือ

b. ค่ากลางระหว่าง max และ min ของ gray level ใน image

2 แบ่ง image เป็น 2 ส่วนโดยใช้ T ดังต่อไปนี้

a. G1 (gray level > T) และ b. G2 (gray level <= T)

3 หาค่าเฉลี่ยของ gray level ของ G1 และ G2 จะได้μ_1 และμ_2ตามลำดับ

4 คำนวณหาค่า threshold ใหม่ T = (μ_1 + μ_2)/2

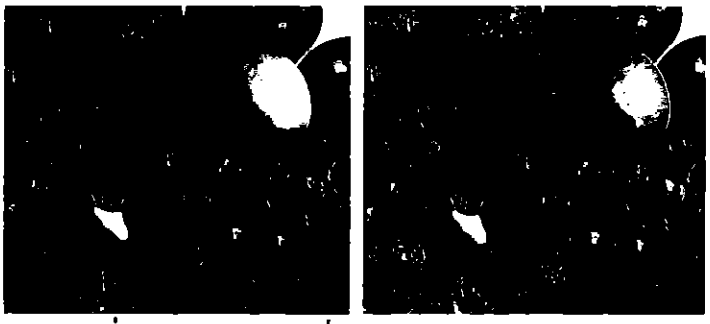
5 ทำซ้ำ ข้อ 2 – 4 จนกระทั่งความแตกต่างระหว่าง T ในรอบถัดกันน้อยกว่า T0 (ค่าที่กำหนดขึ้นมาก่อน)

Automatic Thresholding เริ่มต้นโดยใช้ T = ค่าเฉลี่ยของค่า graylevel และ T0 = 0 ผลลัพธ์คือ T = 125.4 หลังทำซ้ำสามรอบ

รูปที่ 2.4 ตัวอย่าง Automatic Thresholding

2.7 การแยกจุดเด่นของภาพ (ColorIsolator)

เป็นการแยกจุดเด่นออกจากองค์ประกอบโดยรวมของภาพ ซึ่งเป็นวิธีที่ทำให้เกิดการสังเกตเห็นองค์ประกอบนั้นได้ง่าย โดยในโครงงานนี้จะนำมาใช้ในการแยกสีที่ต้องการออกมา จากนั้นใช้ Threshold ในการปรับภาพที่เป็นขาวดำข้างหลังออก จนได้สีที่ต้องการ



รูปที่ 2.5 ตัวอย่างภาพที่ผ่านการทำ ColorIsolation

2.8 โปรแกรม Adobe Flash CS5

โปรแกรม Adobe Flash CS5[8] เป็นโปรแกรมที่มีประสิทธิภาพสูงสำหรับการสร้างกราฟิกแบบเคลื่อนไหวหรือที่เรียกว่าแอนิเมชั่น, เกม ตลอดจนมัลติมีเดียสำหรับเว็บ ซึ่งถูกพัฒนามาอย่างต่อเนื่องตั้งแต่เวอร์ชั่น 3, 4, 5, MX, MX 2004, 8, CS3, CS4 มาจนถึง CS5 ที่เป็นเวอร์ชั่นปัจจุบันในขณะที่ทำโครงงานนี้ ลักษณะเด่นของแอนิเมชั่นที่ได้จากโปรแกรม Flash คือ ไฟล์มีขนาดเล็กจึงสามารถโหลดมาแสดงผลได้อย่างรวดเร็ว รวมทั้งคุณสมบัติในการสร้างภาพกราฟิกแบบเวคเตอร์ทำให้ภาพมีความคมชัด สามารถย่อ – ขยายขนาดได้ โดยมีความละเอียดคงเดิม

ขั้นตอนหลักของการสร้างมัลติมีเดียด้วยโปรแกรม Flash

1. สร้างชิ้นงานภายในโปรแกรม Flash โดยเนื้อหาบางส่วนอาจสร้างขึ้นด้วยเครื่องมือของ Flash เอง และบางส่วนอาจนำเข้า (import) ไฟล์ข้อมูลประเภทต่างๆจากภายนอกเข้ามาใช้
2. ทดสอบการทำงานจนได้ผลที่พอใจ แล้วบันทึกไว้เป็นเอกสารของ Flash (.fla)
3. พับลิชชิ้นงาน (Publish) เป็นไฟล์มูฟวี่ของ Flash (.swf) และหรือไฟล์เว็บเพจ (.html) ซึ่งเป็นข้อมูลที่จะนำไปใช้เผยแพร่

2.8.1 ซิมโบลและอินสแตนซ์

ซิมโบล (Symbol) คือ การนำเอาออบเจ็กต์ที่เป็นภาพกราฟิก ปุ่มกด หรือ มูฟวี่ มากำหนดเป็นต้นแบบ เพื่อนำไปใช้เป็นส่วนประกอบในชิ้นงาน โดยซิมโบลที่ถูกสร้างขึ้นจะถูกจัดเก็บไว้ใน Library เราไม่สามารถนำซิมโบลมาใช้งานบนStage ได้โดยตรง แต่เราจะนำเพียงสำเนาของซิมโบลหรือที่เรียกว่าอินสแตนซ์(Instance)มาใช้ ซึ่งหนึ่งซิมโบลสามารถนำมาใช้งานเป็นอินสแตนซ์ ได้หลายอันบน Stage โดยแต่ละอินสแตนซ์อาจถูกปรับรายละเอียด เช่น ขนาด สี เป็นต้น และการเปลี่ยนแปลงอินสแตนซ์จะไม่มีผลกับซิมโบลแต่การเปลี่ยนแปลงซิมโบลจะมีผลกับอินสแตนซ์ ซิมโบลแบ่งออกเป็นสามชนิดดังนี้

1. Movie clip Symbols เป็นซิมโบลแบบมูฟวี่คลิป ใช้สำหรับการสร้างภาพเคลื่อนไหวแบบสั้นๆ โดยสามารถนำไปใช้ประกอบมูฟวี่ที่สร้างขึ้นได้ตามจำนวนที่ต้องการ
2. Button Symbols เป็นซิมโบลแบบปุ่ม ใช้สำหรับสร้างปุ่มกดเพื่อโต้ตอบกับผู้ใช้โดยผ่านการคลิกเมาส์ วางเมาส์บนปุ่ม หรือคลิกบนปุ่ม สามารถใส่เสียงได้
3. Graphic Symbols เป็นซิมโบลแบบกราฟิกใช้สำหรับการสร้างภาพและส่วนประกอบของภาพเคลื่อนไหว โดยแสดงภาพในบทบาทภาพนิ่ง ไม่สามารถใส่เสียงได้แต่สามารถตอบโต้กับผู้ใช้ได้

2.8.2 ประเภทของไฟล์ Flash

ประเภทของไฟล์ Flash จะเกี่ยวข้องกับไฟล์หลักๆ 2 ประเภทคือ

- ไฟล์เอกสาร (Flash document) มีฟอร์แมตเป็น .fla เป็นไฟล์ที่ได้จากการบันทึกใน

ขั้นตอนการสร้างชิ้นงาน โดยไฟล์จะจัดเก็บข้อมูลต่างๆ ได้จากการออกแบบและพัฒนาทั้งหมดไว้ตามสภาพเดิม และสามมารถเปิดแก้ไขชิ้นงานได้

- ไฟล์มูฟวี่ (Flash movie) มีฟอร์แมตเป็น .swf ไฟล์นี้เป็นผลลัพธ์สุดท้าย ซึ่งจะได้จากการพับลิช (publish) ไฟล์เอกสาร เพื่อนำไปเผยแพร่และแสดงด้วยโปรแกรม Flash Player โดยข้อมูลต้นฉบับจะถูกบีบอัด ปรับลดคุณสมบัติ และยุบรวมกัน เพื่อให้ไฟล์มีขนาดเล็ก ดังนั้นจึงไม่สามารถเปิดขึ้นมาแก้ไขชิ้นงานได้

นอกจากนี้อาจจะมีไฟล์ที่เกี่ยวข้องอีก โดยขึ้นอยู่กับเนื้อหาของชิ้นงานและวิธีการพับลิช รวมไปถึงรูปแบบของการสร้าง เช่น ไฟล์ .flv ซึ่งเป็นไฟล์วีดีโอ (Flash Video) ที่ได้จากการอิมพอร์ตวีดีโอเข้ามาใช้งาน, ไฟล์ .as เป็นไฟล์ที่เก็บซอร์สโค้ดของโปรแกรม ActionScript หรือไฟล์ .html ซึ่งเป็นเว็บเพจที่ใช้แสดงมูฟวี่ เป็นต้น

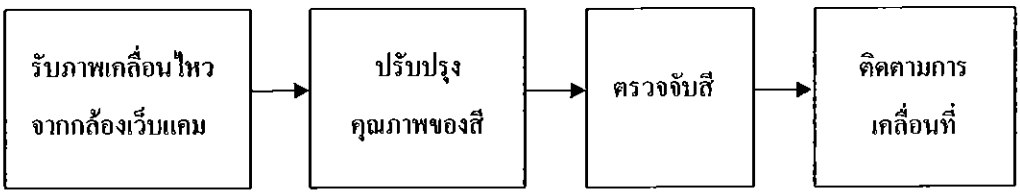
2.9 ภาษา ActionScript 3.0

แอ็กชันสคริปต์ (ActionScript) เป็นภาษาสคริปต์ที่มีพื้นฐานมาจากเอ็กมาสคริปต์ (ECMAScript) ซึ่งเป็นตระกูลเดียวกันกับจาวาสคริปต์ใช้สำหรับการพัฒนาเว็บไซต์หรือซอฟต์แวร์ที่ทำงานด้วยแฟลช เพื่อใช้คำนวณและควบคุมการทำงานในตัวแฟลช และให้สามารถโต้ตอบกับผู้ใช้ได้

2.10 โปรแกรม ColorTracking

โปรแกรม ColorTracking[7] ที่นำมาศึกษาในโครงงานนี้เขียนขึ้นโดย Jonathan Cipriano ซึ่งเป็น creative developer base อยู่ที่ San Francisco โดยโปรแกรมนี้เขียนด้วยภาษา ActionScript 3.0 และให้บุคคลทั่วไปที่สนใจสามารถดาวน์โหลดมาศึกษาได้ จากเว็บไซต์ www.redblind.com ผู้จัดทำโครงงานเล็งเห็นว่าน่าจะนำมาใช้ในการอินเตอร์เฟสกับเกมที่เขียนขึ้นได้ จึงได้นำมาประยุกต์ใช้ในโครงงานนี้

2.10.1 บล็อกไดอะแกรมของโปรแกรม ColorTracking



รูปที่ 2.6 บล็อกไดอะแกรมของโปรแกรม ColorTracking

2.10.2 ส่วนขยายของโปรแกรม ColorTracking

2.10.2.1 การติดต่อกับกล้องเว็บแคม

การติดต่อกับกล้องเว็บแคมในโปรแกรม Flash CS5 จะมีคลาสสำหรับกำหนดให้กล้องเว็บแคมที่เชื่อมต่อกับคอมพิวเตอร์ทำงานเพื่อรับภาพเข้ามาใช้ใน Flash และควบคุมด้วย ActionScript คือ คลาส Camera

คลาส Camera เริ่มต้นโดยการประกาศออบเจ็กต์ที่สร้างเป็น Instance ของคลาสผ่านเมธอด getCamera แทนการใช้ฟังก์ชั่น Construction ซึ่งมีรูปแบบทั่วไปคือ

```
var camName:Camera = Camera.getCamera();
```

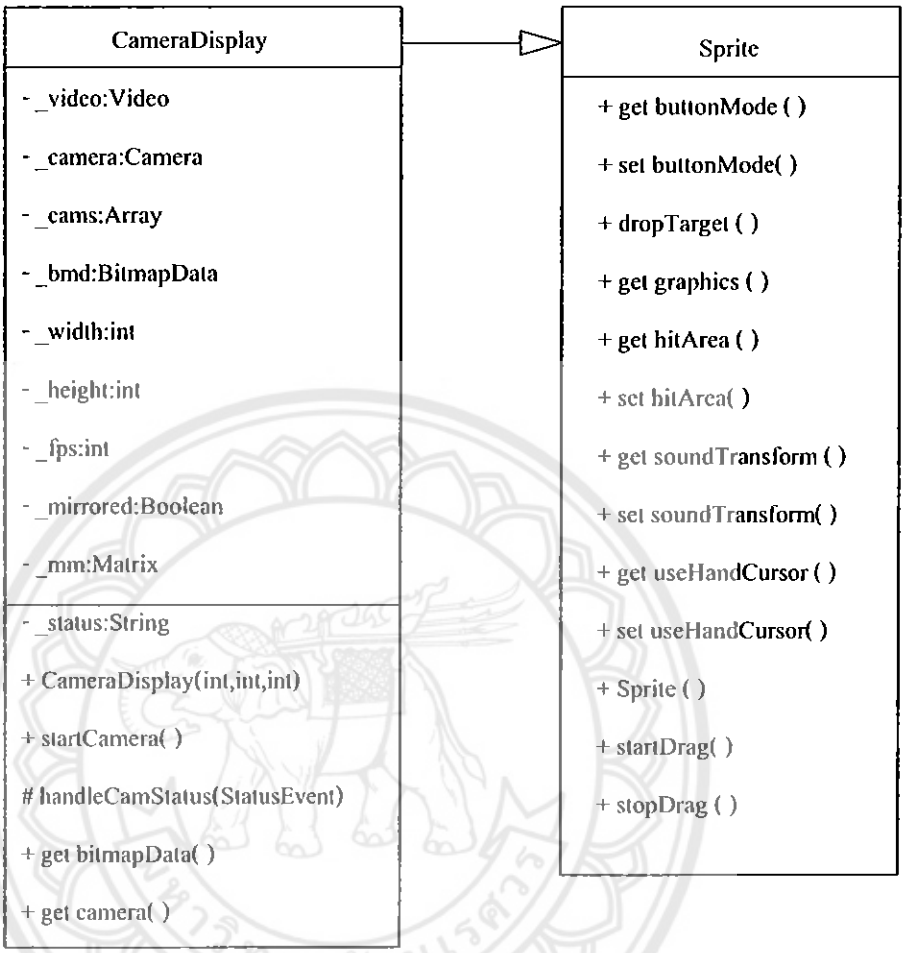
เมธอด getCamera เป็นเมธอดแบบ Static ที่ช่วยในการเชื่อมต่อกล้องเว็บแคมกับคอมพิวเตอร์เพื่อให้ camName อ้างอิงไปยังภาพที่กล้องเว็บแคมกำลังจับภาพอยู่ โดยปรกติเมื่อมีการใช้เมธอดนี้ Flash จะเปิด Privacy Dialog Box ขึ้นมาเพื่อให้ผู้ใช้ทราบว่ามีการใช้งานกล้องเว็บแคมและขออนุญาตจากผู้ใช้ว่าจะยอมให้มีการเชื่อมต่อกับกล้องเว็บแคมหรือไม่ดังรูปที่ 2.7

รูปที่ 2.7 การขออนุญาตเชื่อมต่อกับกล้องเว็บแคม

แต่ Instance ของ Camera ทำหน้าที่เพียงอ้างอิงไปยังกล้องเว็บแคมที่กำลังจับภาพอยู่เท่านั้นไม่ได้มีส่วนแสดงผลเราจึงจำเป็นต้องใช้คลาส Video เพื่อการแสดงผลด้วย ซึ่งการนำมาใช้ร่วมกับ Instance ของ Camera นั้นมีรูปแบบดังนี้

```
var vdoName:Video;
vdoName.attachCamera(camName);
```

2.10.2.2 คลาสไดอะแกรมสำหรับการติดต่อกับกล้องเว็บแคม



รูปที่ 2.8 คลาสไดอะแกรมสำหรับการติดต่อกับกล้องเว็บแคม

โค้ดส่วนของคลาส CameraDisplay (เรียกใช้กล้องจากคอมพิวเตอร์และแสดงผลเป็นภาพวีดีโอ)

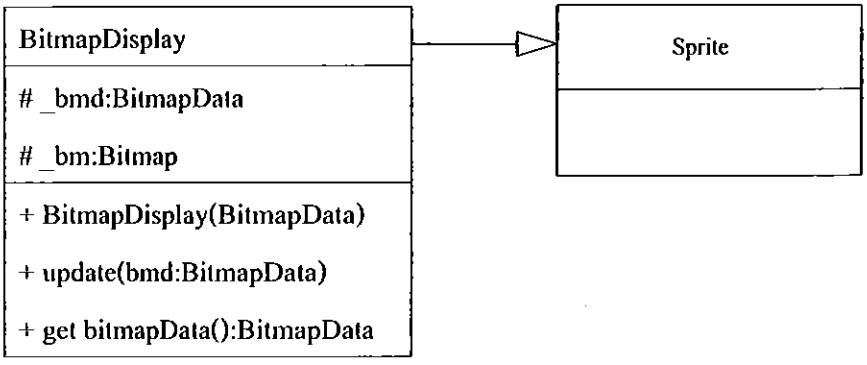
```
package org.redblind.display
    {
        import flash.display.Sprite;
        import flash.events.StatusEvent;
        import flash.geom.Matrix;
        import flash.geom.Point;
        import flash.media.Video;
        import flash.display.DisplayObject;
        import flash.display.BitmapData;
```

```
import flash.media.Camera;

public class CameraDisplay extends Sprite
{
    private var _video:Video;
    private var _camera:Camera;
    private var _cams:Array;
    private var _bmd:BitmapData;

    private var _width:int;
    private var _height:int;
    private var _fps:int;
    private var _mirrored:Boolean;
    private var _mm:Matrix;
    private var _status:String;

    public function CameraDisplay(w:int=320,h:int=240,fps:int=30)
    {
        _video = new Video(w,h);
        _cams = Camera.names;
        trace('CAMERAS: '+_cams);
        _camera = Camera.getCamera();
        _width = w;
        _height = h;
        _fps = fps;
    }

    public function  startCamera():void
    {
        _camera.addEventListener(StatusEvent.STATUS,
        handleCamStatus);
        _camera.setMode(_width,_height,_fps);
```

```
                _video.attachCamera(_camera);
                addChild(_video);
            }
            protected function handleCamStatus(e:StatusEvent):void
            {
                _status = e.code;
            }
            public function get bitmapData():BitmapData
            {
            _bmd = new BitmapData(this.width,this.height,true,0x00FFFFFF);
                if(_mirrored) _bmd.draw(this,_mm);
                else _bmd.draw(this);
                return _bmd;
            }
            public function get camera():Camera { return _camera; }
            public function get status():String { return _status; }
        }
    }
```

จากโค้ดเป็นการติดต่อกับกล้องเว็บแคมและแสดงผลเป็นภาพวีดีโอ นอกจากนี้ยังได้กำหนดขนาดของกล้องเว็บแคมและแปลงภาพวีดีโอให้อยู่ในรูปบิตแมปอีกด้วยซึ่งการแสดงผลภาพบิตแมปมีการเขียนคลาสเพิ่มชื่อว่า BitmapDisplay ดังคลาสไดอะแกรมต่อไปนี้

2.10.2.3 คลาสไดอะแกรมสำหรับการแสดงผลวีดีโอเป็นแบบบิตแมป



รูปที่ 2.9 คลาสไดอะแกรมสำหรับการแสดงผลวีดีโอแบบบิตแมป

```
package org.redblind.display

{
    import flash.display.Sprite;
    import flash.display.Bitmap;
    import flash.display.BitmapData;
    /**
    * @author jcipriano
     */
    public class BitmapDisplay extends Sprite

    {
        protected var _bmd:BitmapData;
        protected var _bm:Bitmap;
        public function BitmapDisplay(bmd:BitmapData=null):void

        {
            if(bmd){
                _bmd = bmd;
                _bm = new Bitmap(_bmd);
                addChild(_bm);
            }
        }
        public function update(bmd:BitmapData):void

        {
            _bmd = bmd;
            if(_bm){
                removeChild(_bm);
                _bm = null;
            }
            _bm = new Bitmap(_bmd);
```

```
                addChild(_bm);
            }
            public function get bitmapData():BitmapData

            {
                return _bm.bitmapData;
            }
        }
    }
```

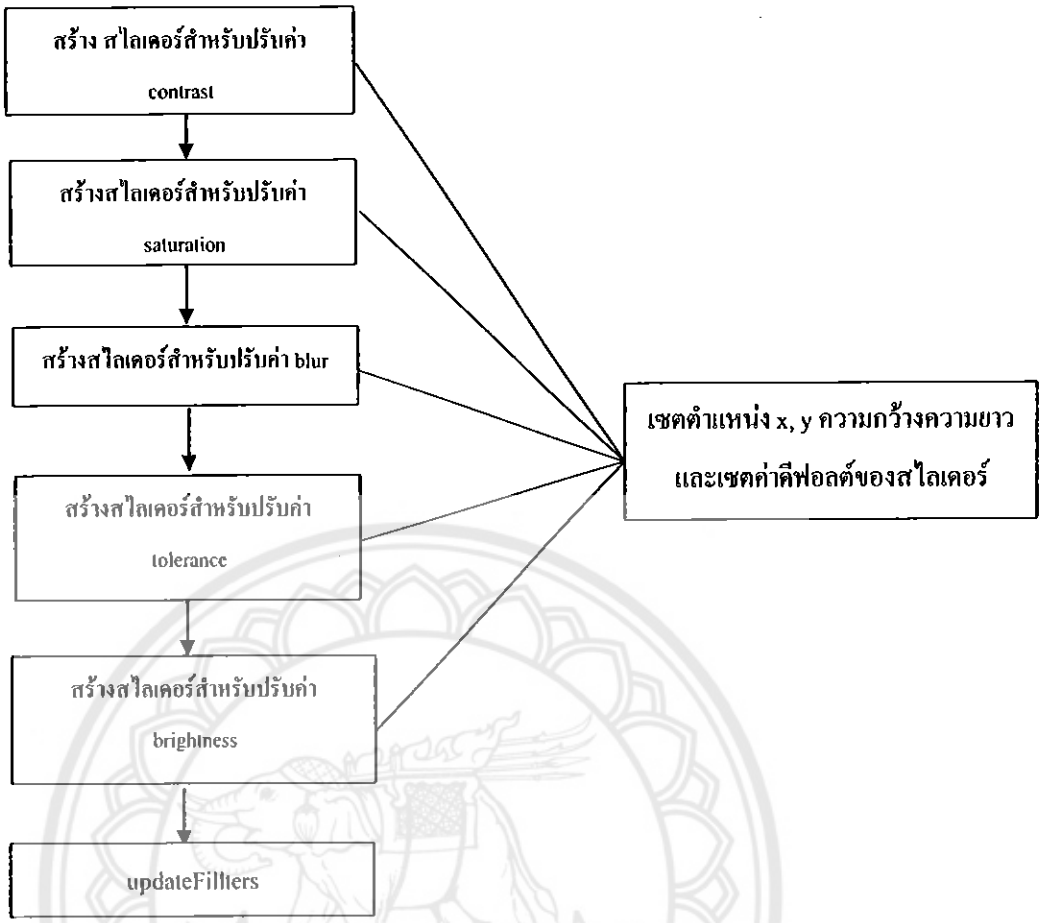
2.10.2.4 ปรับปรุงคุณภาพของสี (Color Enhancement)

ในขั้นตอนนี้เป็นการเพิ่มความคมชัดของสีโดยการปรับค่าต่างๆคือ contrast, saturation, blur, tolerance, brightness โดยใช้ไลบารี่ bit101 ในการทำสไลเดอร์สำหรับปรับค่า

Contrast	คือค่าความเปรียบต่างระหว่างส่วนที่มืดที่สุดกับส่วนที่สว่างที่สุด
Saturation	คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย
Blur	การปรับค่าความคมชัดของสีภาพ
Tolerance	คือค่าที่กำหนดในการพิจารณาเลือกพื้นที่ สามารถกำหนดค่าได้ตั้งแต่ 0-255 ถ้าหากค่าที่กำหนดมีค่าน้อย จะเลือกพื้นที่จากสีที่ใกล้เคียงและเลือกพื้นที่ได้แคบ แต่ถ้ากำหนดค่าสูงจะเลือกพื้นที่ได้กว้างขึ้นโดยพิจารณาจากสีที่ใกล้เคียงกับสีที่กำหนด
Brightness	ค่าความมืดความสว่างของภาพ

ขั้นตอนการปรับปรุงคุณภาพสีมีดังต่อไปนี้

1. การสร้างสไลเดอร์ในโปรแกรม ColorTracking เป็นการเลือกใช้ HSlider (Horizontal Slider) เพื่อใช้ปรับค่า ซึ่งมีการจัดตำแหน่ง,ขนาดและเซตค่าดีฟอลต์ของสไลเดอร์ ซึ่งค่าทั้งหมดจะถูกอัพเดทเพื่อแสดงผลในภาพวีดีโอ แสดงได้ดังโฟว์ชาร์ตต่อไปนี้



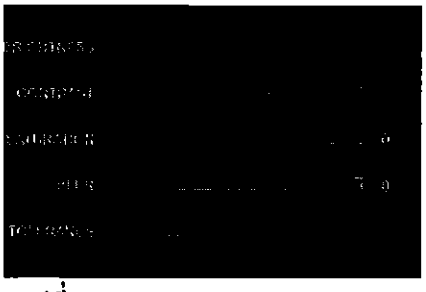
รูปที่ 2.10 แสดงการสร้างสไลเดอร์ในการปรับภาพในฟังก์ชั่น setupControl

การสร้างสไลเดอร์อยู่ในฟังก์ชั่น setupControl ภายในคลาส ColorTracking มีการเขียนโค้ดดังนี้

```
private function setupControls():void
        {
                contrast = new HSlider(this,0,0,updateFilters);
                contrast.width = sliderWidth;
                contrast.height = 10;
                contrast.x = contrast_txt.x - contrast.width - 10;
                contrast.y = contrast_txt.y + 1;
                contrast.setSliderParams(0,100,75);//contrast = 75
                addChild(contrast);
                saturation = new HSlider(this,0,0,updateFilters);
                saturation.width = sliderWidth;
                saturation.height = contrast.height;
                saturation.x = saturation_txt.x - saturation.width - 10;
```

```
        saturation.y = saturation_txt.y + 1;
        saturation.setSliderParams(0,100,50);//saturation = 50
        addChild(saturation);

        blur = new HSlider(this,0,0,updateFilters);
        blur.width = sliderWidth;
        blur.height = contrast.height;
        blur.x = blur_txt.x - blur.width - 10;
        blur.y = blur_txt.y + 1;
        blur.setSliderParams(0,10,10);//blur = 10
        addChild(blur);

        tolerance = new HSlider(this,0,0,updateFilters);
        tolerance.width = sliderWidth;
        tolerance.height = contrast.height;
        tolerance.x = tolerance_txt.x - tolerance.width - 10;
        tolerance.y = tolerance_txt.y + 1;
        tolerance.setSliderParams(0,200,5);//ค่า tolerance = 5
        addChild(tolerance);
        brightness = new HSlider(this,0,0,updateFilters);
        brightness.width = sliderWidth;
        brightness.height = contrast.height;
        brightness.x = brightness_txt.x - brightness.width - 10;
        brightness.y = brightness_txt.y + 1;
        brightness.setSliderParams(0,100,15); //ค่า brightness = 15
        addChild(brightness);
        updateFilters();
    }
```

จากโค้ดด้านบนเป็นการเรียกใช้งาน HSlider ซึ่งเป็นสไลเดอร์แนวนอนรวมไปถึงการจัดตำแหน่งของสไลเดอร์ทั้งหมดและการเซตค่าเริ่มต้นให้กับสไลเดอร์ซึ่งจะแสดงผลดังรูปที่ 2.11

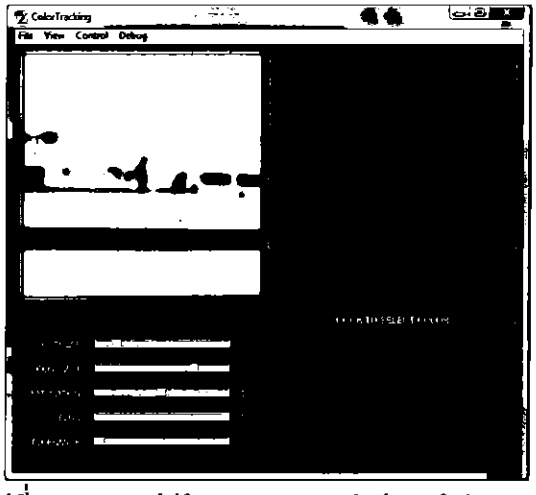


รูปที่ 2.11 UI สำหรับปรับปรุงภาพ

2. อัพเดทฟิลเตอร์

ในส่วนนี้เป็นการอัพเดทค่าต่างๆที่เซตไว้จากขั้นตอนแรกในขั้นตอนนี้ภาพวีดีโอจะถูกปรับปรุงให้มีค่าตามที่ตั้งไว้เป็นค่าเริ่มต้น ภาพวีดีโอในขั้นตอนนี้จะเป็นดังรูปที่ 2.12

```
private function updateFilters(e:Event=null):void

{
        brightness_txt.text = Math.floor(brightness.value).toString();
        saturation_txt.text = Math.floor(saturation.value).toString();
        contrast_txt.text = Math.floor(contrast.value).toString();
        blur_txt.text = Math.floor(blur.value).toString();
        tolerance_txt.text = Math.floor(tolerance.value).toString();

        cm = new ColorMatrix();

        cm.adjustContrast(contrast.value);
        cm.adjustSaturation(saturation.value);
        cm.adjustBrightness(brightness.value);

        camDisp.filters = [new ColorMatrixFilter(cm),
                new BlurFilter(blur.value,blur.value,1)];

        ci.tolerance = tolerance.value;
}
```



รูปที่ 2.12 ภาพวีดีโอจากการเซตค่าสีของโปรแกรม

การปรับค่าต่างๆที่กล่าวมามีการคำนวณโดยคลาส ColorMatrix แสดงดังโค้ดต่อไปนี้

```
public function adjustColor(p_brightness:Number,p_contrast:Number,
                            p_saturation:Number, p_hue:Number):void {
                            adjustHue(p_hue);
                            adjustContrast(p_contrast);
                            adjustBrightness(p_brightness);
                            adjustSaturation(p_saturation);
}
```

จากโค้ดเป็นฟังก์ชั่นการรับค่าต่างๆมาจากฟังก์ชั่น adjustHue, adjustContrast, adjustBrightness, adjustSaturation ซึ่งเป็นฟังก์ชั่นที่กำหนดค่าต่างๆเวลาเลื่อนสไลเดอร์

ฟังก์ชั่น adjustHue

เป็นฟังก์ชั่นการปรับค่า hue ซึ่งมีการกำหนดไว้ว่าค่าที่ปรับจะต้องไม่เกิน 180 และมีค่าความสว่างของสีแดงเท่ากับ 0.213, สีเขียวเท่ากับ 0.715, สีน้ำเงินเท่ากับ 0.072 ซึ่งถ้าหากเราปรับค่าจากสไลเดอร์ ค่า sin, cos ของ hue จะเปลี่ยนไปตามค่าที่ปรับจากฟังก์ชั่น Math.cos (p_val), Math.sin(p_val) ค่า p_val ค่าค่าที่ได้รับมาจากการปรับและเมตริกซ์ 5x5 ของค่า hue จะเป็นดังนี้

```
[
lumR+cosVal*(1-lumR)+sinVal*(-lumR),lumG+cosVal*(-lumG)+sinVal*(-lumG),
lumB+cosVal*(-lumB)+sinVal*(1-lumB),0,0,
lumR+cosVal*(-lumR)+sinVal*(0.143),lumG+cosVal*(1-lumG)+sinVal*(0.140),
```

lumB+cosVal*(-lumB)+sinVal*(-0.283),0,0,

lumR+cosVal*(-lumR)+sinVal*(-(1-lumR)),lumG+cosVal*(-lumG)+sinVal*(lumG),

lumB+cosVal*(1-lumB)+sinVal*(lumB),0,0,

0,0,0,1,0,

0,0,0,0,1

]

ค่า lumR คือ ค่าความสว่างของสีแดง

ค่า lumG คือ ค่าความสว่างของสีเขียว

ค่า lumB คือ ค่าความสว่างของสีน้ำเงิน

ฟังก์ชั่น adjustContrast

เป็นฟังก์ชั่นการปรับค่า contrast ซึ่งกำหนดตัวแปรสำหรับรับค่าเป็น p_val และค่าที่สไลเดอร์ปรับได้จะต้องไม่เกิน 100 เมื่อมีการปรับค่าที่สไลเดอร์ค่า p_value ก็จะเปลี่ยนไปตามค่าที่ปรับ และค่า contrast มี colorMatrix ดังนี้

[

x/127,0,0,0,0.5*(127-x),

0,x/127,0,0,0.5*(127-x),

0,0,x/127,0,0.5*(127-x),

0,0,0,1,0,

0,0,0,0,1

]

ค่า x คือค่าที่ได้จากค่า p_val mod 1 หรือ x = 127 +p_val/100*127 (เมื่อ p_val<0)

ฟังก์ชั่น adjustBrightness

เป็นฟังก์ชั่นการปรับค่า brightness ซึ่งซึ่งกำหนดตัวแปรสำหรับรับค่าเป็น p_val และค่าที่สไลเดอร์ปรับได้จะต้องไม่เกิน 100 มีเมตริกซ์ 5x5 ดังนี้

[

1,0,0,0,p_val,

0,1,0,0,p_val,

0,0,1,0,p_val,

```
        0,0,0,1,0,
        0,0,0,0,1
    ]
```

ฟังก์ชั่น adjustSaturation

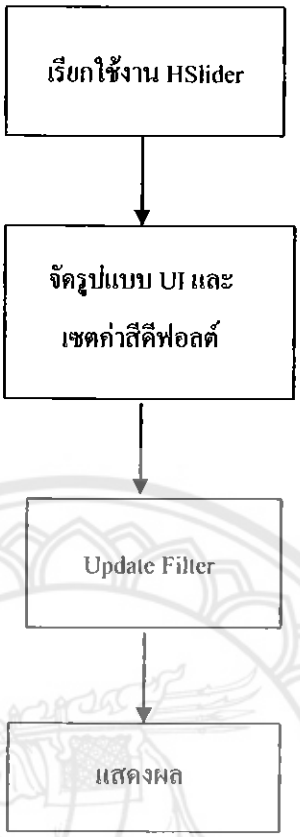
เป็นฟังก์ชั่นการปรับค่า saturation ซึ่งมีการกำหนดไว้ว่าค่าที่ปรับจะต้องไม่เกิน 100 และมีค่าความสว่างของสีแดงเท่ากับ 0.3086, สีเขียวเท่ากับ 0.6049, สีน้ำเงินเท่ากับ 0.0820 ซึ่งถ้าหากเราเมตริกซ์ 5x5 ของค่า saturation จะเป็นดังนี้

```
[
    lumR*(1-x)+x,lumG*(1-x),lumB*(1-x),0,0,
    lumR*(1-x),lumG*(1-x)+x,lumB*(1-x),0,0,
    lumR*(1-x),lumG*(1-x),lumB*(1-x)+x,0,0,
    0,0,0,1,0,
    0,0,0,0,1
]
```

ค่า lumR คือ ค่าความสว่างของสีแดง
ค่า lumG คือ ค่าความสว่างของสีเขียว
ค่า lumB คือ ค่าความสว่างของสีน้ำเงิน
ค่า x = 1+(3*p_val/100) ; ถ้า p_val > 0
ค่า x = p_val/100 ; ถ้า p_val < 0

2.10.2.5 สรุปขั้นตอนการปรับปรุงคุณภาพของสี

สำหรับการปรับปรุงคุณภาพสีในโปรแกรม ColorTracking จะเริ่มด้วยการเรียกใช้ HSlider ซึ่งเป็นสไลเดอร์แนวนอนจากไลบรารี bit101 ก่อนจากนั้นจะเป็นการเขียนโค้ดเพื่อจัดรูปแบบสไลเดอร์และกำหนดค่าดีฟอลต์ต่างๆ ซึ่งการกระทำทั้งหมดที่กล่าวมานี้จะเขียนไว้ในฟังก์ชั่น setupControls() ค่าที่เซตไว้จะถูกอัพเดทไปยังคลาส ColorMatrix และส่งผลไปยังการแสดงผลภาพวีดีโอ แสดงดังโฟว์ชาร์ตต่อไปนี้



รูปที่ 2.13 สรุปขั้นตอนการปรับปรุงคุณภาพของสี

2.10.2.6 การตรวจจับสี

โปรแกรม ColorTracking มีคลาสที่ชื่อว่า ColorIsolator เป็นคลาสสำหรับการแยกสี โดยการใช้ เมธอด threshold( ) ซึ่งวิธีการที่ทำให้สามารถแยกสีและแทนที่ช่วงของสีในภาพได้และเป็นเมธอดหนึ่งในคลาส BitmapData มีการเรียกใช้ดังนี้

```
public function threshold (sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point,
                           operation:String, threshold:uint, color:uint = 0, mask:uint =
                           4294967295, copySource:Boolean = false) : uint;
```

จากฟังก์ชั่นด้านบนสามารถอธิบายได้ดังนี้

sourceBitmapData คืออินพุตรูปบิตแมปที่จะใช้ ภาพจะแตกต่างกับออบเจ็กต์ BitmapData หรือสามารถอ้างถึง BitmapData ที่มีอยู่ได้

sourceRect คือรูปสี่เหลี่ยมที่ใช้กำหนดพื้นที่ของภาพที่เป็นอินพุท

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

destPoint	จุดเป้าหมายของภาพ(BitmapDataในขณะนั้น)ซึ่งจะสอดคล้องกับมุมบนซ้ายของสี่เหลี่ยม
operation	สัญลักษณ์เปรียบเทียบอย่างใดอย่างหนึ่ง ต่อไปนี้ในรูปแบบ String "<", "<=", ">", ">=", "==", "!="
threshold	ค่าที่มีการทดสอบแต่ละพิกเซลเพื่อดูว่าจะตรงหรือเกินกว่าค่า threshold
color	ค่าพิกเซลของสีจะถูกตั้งค่าถ้าการทดสอบ threshold สำเร็จ ค่าเริ่มต้นคือ 0x00000000.
mask	ใช้เพื่อแยกส่วนประกอบของสี
copySource	ถ้ามีค่าเป็น true ค่าสีจากรูปภาพต้นทางจะถูกคัดลอกไปยังปลายทางเมื่อการทดสอบ threshold ผิดพลาด ถ้ามีค่าเป็น false ภาพต้นทางจะไม่ถูกคัดลอกเมื่อการทดสอบ threshold ผิดพลาด

บางส่วนของคลาส ColorIsolator ที่มีการใช้เมธอด Threshold

```
_bmd.threshold(_bmd, _bmd.rect, _pt, "<", rmin<<16,0,0x00FF0000,true);
_bmd.threshold(_bmd, _bmd.rect, _pt, "<", gmin << 8,0,0x0000FF00,true);
_bmd.threshold(_bmd, _bmd.rect, _pt, "<", bmin,0,0x000000FF,true);
_bmd.threshold(_bmd, _bmd.rect, _pt, ">", rmax << 16,0,0x00FF0000,true);
_bmd.threshold(_bmd, _bmd.rect, _pt, ">", gmax << 8,0,0x0000FF00,true);
_bmd.threshold(_bmd, _bmd.rect, _pt, ">", bmax,0,0x000000FF,true);
```

เมื่อมีการทำใช้เมธอด threshold ในการแยกค่าสีจะมีผลดังรูปที่ 2.14

รูปที่ 2.14 ผลที่ได้หลังจากการใช้เมธอด threshold

1573471X
ม/ร.
ว 4149
2553

จากรูปที่ 2.14 ภาพด้านซ้ายเป็นภาพที่ปรับความคมชัดของสีแล้ว เมื่อคลิกเลือกสีฟ้า จะได้ผลลัพธ์ดังจอด้านขวามือที่เป็นผลมาจากการใช้ threshold

2.10.2.7 กำหนดขอบเขตพื้นที่ของสีโดยใช้ getColorBoundRect

เมธอด getColorBoundRect เป็นเมธอดที่ใช้ในการกำหนดพื้นที่สี่เหลี่ยมให้ล้อมรอบสีที่ระบุไว้ภายในรูปแบบบิตแมป(ถ้าพารามิเตอร์ findColor ถูกตั้งค่าเป็น true) หรือกำหนดให้ล้อมรอบสีทั้งหมดที่ไม่ใช่สีที่ระบุไว้(ถ้าพารามิเตอร์ findColor ถูกตั้งค่าเป็น false) ซึ่งเมธอดนี้อยู่ในคลาส BitmapData เช่นเดียวกันกับเมธอด threshold มีการเรียกใช้ดังนี้

```
public function getColorBoundsRect (mask:uint, color:uint, findColor:Boolean =
true) : Rectangle;
```

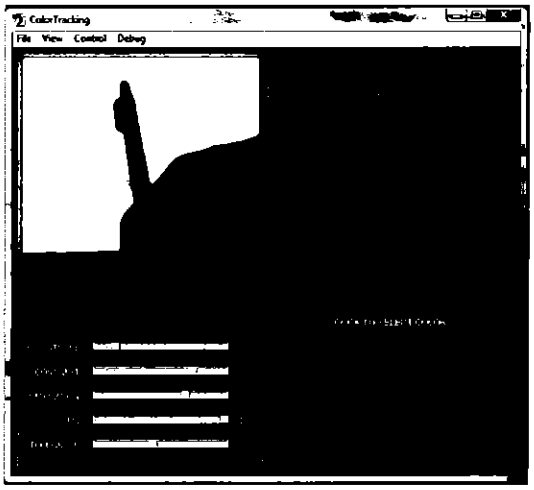
จากฟังก์ชั่นด้านบนอธิบายได้ดังนี้

mask	ค่าตัวเลขฐานสิบหกที่เป็นการกำหนดบิตของสี ARGB ที่จะต้องพิจารณา ค่าของสีจะรวมกับค่าตัวเลขฐานสิบหกนี้โดยใช้เครื่องหมาย & (AND)
color	เป็นค่าตัวเลขฐานสิบหก เป็นการกำหนดสีเพื่อให้ตรงกับ ARGB (ถ้า findColor ถูกตั้งค่าเป็น true) หรือเพื่อไม่ให้ตรงกันกับ ARGB (ถ้า findColor ถูกตั้งค่าเป็น false)
findColor	ถ้าตั้งค่าเป็น true จะรีเทรินค่าขอบเขตของสีในภาพ ถ้าตั้งค่าเป็น false จะรีเทรินขอบเขตของสีที่ไม่ได้อยู่ในภาพ
return	พื้นที่ของภาพตามที่ได้ระบุสีไว้

เมธอดนี้ถูกเรียกใช้ในคลาส ColorIsolator โดยมีการเรียกใช้งานดังนี้

```
_rect = _bmd.getColorBoundsRect(0xFFFFFFFF, _color, false);
//_color คือค่าที่ได้จากฟังก์ชั่น setColor ในคลาส ColorTracking
```

เมื่อใช้ getColorBoundRect จะมีกรอบสี่เหลี่ยมมาล้อมรอบสีที่เลือกดังรูปที่ 2.15



รูปที่ 2.15 หลังจากใช้เมธอด getColorBoundsRect ติดตามสีที่ต้องการ

จากรูปที่ 2.15 จอภาพด้านขวามือเป็นการบอกพิกัดของสีที่เลือกโดยมีกรอบสี่เหลี่ยมล้อมรอบสีนั้นเมื่อมีการเคลื่อนที่กรอบสี่เหลี่ยมก็จะเคลื่อนที่ตาม

2.11 การแปลงรูปภาพระดับพิกเซล (BitmapData Class)

BitmapData เป็นคลาสๆหนึ่งในโปรแกรม Flash ใช้สำหรับแปลงรูปภาพจากก้อนใหญ่ๆ เป็น pixel เล็กๆหลายๆอันมาต่อกัน ใช้ในการดัดแปลงรูปภาพระดับพิกเซล และการตรวจจับการชนในระดับพิกเซล ซึ่งในการทำ BitmapData นั้นจะต้องใช้ Color Matrix มาทำการแยกแยะแสงสีต่างๆซึ่งมีการคำนวณมาแล้วใน Class ColorMatrix ซึ่งเป็น Library ที่เขียนขึ้นโดย Grant Skinner

2.12 การจัดการเกี่ยวกับสี (ColorMatrix)

ColorMatrix คือ ตาราง 2 มิติขนาด 5x5 มีไว้สำหรับจัดการเกี่ยวกับสี โดยเรียงลำดับสมาชิกตั้งแต่ 0 - 4 ทั้งแนวนอน (row) และแนวตั้ง (column)

แถวที่ 1 จัดการกับสีแดง (R)
แถวที่ 2 จัดการกับสีเขียว (G)
แถวที่ 3 จัดการกับสีน้ำเงิน (B)
แถวที่ 4 จัดการกับความโปร่งใสของแสง (A)
แถวที่ 5 สำหรับเพิ่มลดค่าของแถวด้านบน (w)

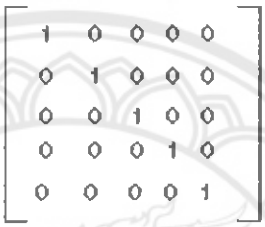
คอลัมน์ที่ 1 จัดการกับสีแดง (R)
คอลัมน์ที่ 2 จัดการกับสีเขียว (G)

คอลัมน์ที่ 3 จัดการกับสีน้ำเงิน (B)

คอลัมน์ที่ 4 จัดการกับแสง (ได้เฉพาะแถว 4 และ 5 เท่านั้น)

คอลัมน์ที่ 5 ไม่มีการจัดการ

ตารางมาตรฐานของ ColorMatrix จะมีค่าเมนหลักของแต่ละตัวเป็น 1 ในแต่ละแถวคอลัมน์ [0][0], [1][1], [2][2], [3][3], [4][4] เรียงตามลำดับในการจัดการกับ สีแดง, สีเขียว, สีน้ำเงิน, ความโปร่งใสของแสง, (ส่วนสุดท้ายจะต้องกำหนดเป็น 1 ตลอดไม่มีการเปลี่ยนแปลง) ส่วนสมาชิกตัวอื่นๆ จะเป็น 0 ทั้งหมด



รูปที่ 2.16 ค่าดีฟอลต์ของ ColorMatrix

การจัดการกับ 3 แถวแรก แดง เขียว น้ำเงิน จะกำหนดค่าการเปลี่ยนแปลงเพียง 3 คอลัมน์แรก คือ [x][0] , [x][1] , [x][2]

การจัดการกับแถวที่ 4 ความโปร่งใสของแสง จะกำหนดค่าการเปลี่ยนแปลงเพียง คอลัมน์ที่ 4 คือ [3][3] เท่านั้น

การจัดการกับแถวที่ 5 กำหนดค่าได้ 4 คอลัมน์ [4][0] , [4][1] , [4][2] , [4][3]

2.12.1 การแปลงสีให้เป็นระบบเมตริกซ์

แบ่งสีออกเป็น 5 คอลัมน์ ตามลักษณะของเมตริกซ์

[แดง เขียว น้ำเงิน แสง 1.0]

[0.2 0.0 0.4 1.0 1.0]

ซึ่งสมาชิกทั้งหมดนี้จะมีค่าระหว่าง 0.0 ถึง 1.0 เท่านั้น ส่วนค่าทั้งหมดหาได้จาก

สี RGB (0-255) หาร 255

ความโปร่งใส (0-100) หาร 100

ค่าสุดท้ายเป็น 1.0 ตลอด

$$\begin{bmatrix} 0.2 & 0.0 & 0.4 & 1.0 & 1.0 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0.2 & 0.2 & 0.2 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.6 & 0.2 & 0.6 & 1.0 & 1.0 \end{bmatrix}$$

รูปที่ 2.17 การคำนวณด้วย ColorMatrix

2.13 แปลงค่าเมตริกซ์ (ColorMatrixFilter)

เป็นคลาสสำหรับการแปลงค่าเมตริกซ์ให้เป็นขนาด 4 x 5 ในสี RGBA และค่าอัลฟ่าของทุกๆพิกเซลในภาพอินพุท เพื่อสร้างผลลัพธ์ให้กับสี RGBA เซตใหม่ และค่าอัลฟ่า

เป็นการช่วยให้เกิดการเปลี่ยนแปลงของค่า saturation, ค่า hue, ความโปร่งแสงของค่าอัลฟ่า เป็นต้น ซึ่งสามารถนำไปใช้ได้กับ display object เช่น MovieClip, SimpleButton, TextField และ Video object เช่นเดียวกันกับ Bitmapdata object

ค่า RGBA ค่าไบต์ที่สำคัญที่สุดจะแสดงในช่วงสีแดง ตามด้วย เขียว, น้ำเงิน, และอัลฟ่า ตามลำดับ การสร้าง color matrix filter ขึ้นมาใหม่นั้นจะใช้คำสั่ง new ColorMatrixFilter() การเซตค่าฟิลเตอร์จะไม่มีผลใดๆกับออบเจ็กต์ที่นำมาใช้ สามารถทำให้กลับไปเป็นเหมือนเดิมได้

การแปลงค่า (ให้สีแดง, เขียว, น้ำเงิน และอัลฟ่าเท่ากับ srcR, srcG, srcB, srcB) เมตริกซ์ขนาด 4 x 5 ที่มีจำนวนอาร์เรย์เท่ากับ 20 ค่า หาได้ดังนี้

redResult = (a[0] ~~ srcR) + (a[1] ~~ srcG) + (a[2] ~~ srcB) + (a[3] ~~ srcA) + a[4]

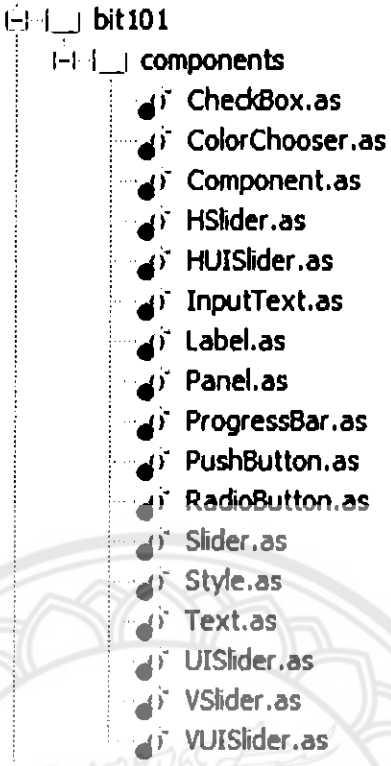
greenResult = (a[5] ~~ srcR) + (a[6] ~~ srcG) + (a[7] ~~ srcB) + (a[8] ~~ srcA) + a[9]

blueResult = (a[10] ~~ srcR) + (a[11] ~~ srcG) + (a[12] ~~ srcB) + (a[13] ~~ srcA) + a[14]

alphaResult = (a[15] ~~ srcR) + (a[16] ~~ srcG) + (a[17] ~~ srcB) + (a[18] ~~ srcA) + a[19]

2.14 ไลบรารี่ที่ใช้สำหรับการทำ UI (Library bit101)

เป็นไลบรารีที่ใช้สำหรับการทำ UI ในโปรแกรม Flash ซึ่งเขียนด้วยภาษา Action Script 3.0 โดย Keith Peters เมื่อเรียกใช้คลาสที่อยู่ในไลบรารีจะเป็นการสร้าง UI ขึ้นมาสำหรับให้ใช้งานได้สะดวกเช่น สไลเดอร์สำหรับปรับสี เป็นต้น มีส่วนประกอบต่างๆดังรูปที่ 2.18



รูปที่ 2.18 ส่วนประกอบต่างๆของ Library bit101

2.15 การทำให้ออบเจ็กต์มีการเปลี่ยนแปลง (Tweening)

Tweening คือการทำให้ออบเจ็กต์มีการเปลี่ยนแปลง เช่น เคลื่อนที่, หมุน, สว่างขึ้น, มืดลง, หรือจางหายไป เป็นต้น การทำ Tweening มี 2 แบบดังนี้

1. Motion Tween เป็นการสร้างภาพเคลื่อนไหวที่มีการกำหนดการเคลื่อนที่ หมุน ย่อ หรือขยายให้ออบเจ็กต์ เป็นรูปแบบการสร้างภาพเคลื่อนไหวที่ใช้มากที่สุด และโปรแกรมจะกำหนดรูปแบบชนิดนี้เป็นค่าเริ่มต้นของการสร้างภาพเคลื่อนไหวทุกครั้ง โดยโปรแกรมจะสร้างภาพเคลื่อนไหวและเส้นทางการเคลื่อนไหวให้เองโดยอัตโนมัติ

2. Shape Tween เป็นการสร้างภาพเคลื่อนไหวโดยการเปลี่ยนแปลงรูปทรงของออบเจ็กต์จากรูปทรงหนึ่งไปเป็นอีกรูปทรงหนึ่งโดยสามารถกำหนด ทิศทาง ตำแหน่ง ขนาด และสีของการเปลี่ยนแปลงได้ตามต้องการ

2.16 ไลบรารีสำหรับการทำให้วัตถุเคลื่อนที่ตามเป้าหมาย (Library TweenLite)

TweenLite เป็นเครื่องมือในการทำ Tweening ที่มีขนาดเล็กมาก, มีความรวดเร็วและยืดหยุ่น เป็นเครื่องมือหลักที่ให้ใช้บริการของแพลตฟอร์ม tweening GreenShock

มีการเรียกใช้ดังนี้

TweenLite($target:Object, $duration:Number, $vars:Object)

พารามิเตอร์ต่างๆอธิบายได้ดังต่อไปนี้

1. $target : Object - ออบเจ็กต์เป้าหมายที่มีคุณสมบัติตามที่กำลังจะ Tweenning

2. $duration : Number - ระยะเวลา (วินาที) ของการ Tween

3. $vars : Object - ค่าสิ้นสุดออบเจ็กต์ของคุณสมบัติทั้งหมดที่ต้องการ tween (หรือถ้าใช้ TweenLite.from() ตัวแปรนี้จะเป็นการกำหนดค่าเริ่มต้น)

ตัวอย่างการใช้ TweenLite

TweenLite.to(myClip, 1, {alpha:0.5})//tween ค่าแอลฟ่าของ myClip เป็น 0.5 ภายในเวลา 1 วินาที

การอิมพอร์ตมาใช้งาน

```
Import com.greenshock.TweenLite;
```

บทที่ 3
รายละเอียดการดำเนินงาน

โครงงานนี้ได้ศึกษาอัลกอริทึมสำหรับการตรวจจับและติดตามการเคลื่อนที่ของมือเพื่อใช้ในการอินเตอร์เฟสกับเกม ซึ่งได้ศึกษาจากโปรแกรม ColorTracking ที่เป็นโปรแกรมสำหรับตรวจจับและติดตามสี ผู้จัดทำได้นำอัลกอริทึมของโปรแกรมมาประยุกต์ใช้ในโครงงาน โปรแกรม ColorTracking นี้จะมีสไลเดอร์สำหรับปรับปรุงสีและมีปุ่มสำหรับเลือกสีที่ต้องการติดตาม ซึ่งในส่วนนี้จะต้องมีการออกแบบใหม่ซึ่งจะกล่าวต่อไป ในการดำเนินงานจะเริ่มจาก ออกแบบเกมให้เหมาะสมกับอัลกอริทึมที่ใช้ และเขียนโปรแกรมควบคุมการกระทำต่างๆของเกม จากนั้นปรับปรุงโปรแกรม ColorTracking ให้สามารถนำมาใช้กับการอินเตอร์เฟสกับเกมเพื่อควบคุมตัวละครและส่วนประกอบต่างๆของเกมที่ออกแบบไว้ได้ โดยที่ผู้ใช้โปรแกรมหรือผู้เล่นเกมไม่ต้องปรับค่าใดๆทั้งสิ้น เมื่อกล้องเว็บแคมทำงานก็สามารถติดตามการเคลื่อนที่ของมือได้ทันที

3.1 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงานในการนำหลักการประมวลผลภาพวีดีโอมาใช้ในการอินเตอร์เฟสกับเกมมีดังต่อไปนี้

1. ออกแบบเกม ในขั้นตอนนี้เป็นการออกแบบตัวละคร, ฉากที่ใช้ในเกม และการเขียนโปรแกรมเพื่อควบคุมแอ็คชั่นต่างๆของตัวละครแต่ละตัว และเขียนโปรแกรมการควบคุมเกมทั้งหมดให้เป็นไปตามเงื่อนไขที่กำหนด

2. ปรับปรุงโปรแกรม ColorTracking ให้เหมาะสมกับเกมที่ออกแบบไว้ และอินเตอร์เฟสกับเกม เป็นการนำเอาอัลกอริทึมของโปรแกรม ColorTracking มาประยุกต์ใช้ และปรับปรุงส่วนประกอบต่างๆของจอแสดงผลเพื่อให้เป็นไปตามเกมที่ออกแบบไว้

3. Publish โปรแกรม หลังจากปรับโครงสร้างต่างๆของโปรแกรมและเขียนโปรแกรมเรียบร้อยแล้ว จะต้องพับลิชโปรแกรมเพื่อนำมาใช้งาน ซึ่งไฟล์ที่ได้จากการพับลิช คือไฟล์ .swf ที่สามารถนำมาใช้งานได้โดยไม่ต้องมีการติดตั้ง

ในการดำเนินงานในแต่ละขั้นตอนประกอบไปด้วยการทำงานในหลายๆส่วนซึ่งมีรายละเอียดดังต่อไปนี้

3.2 ออกแบบเกม

หลักการทำงานโดยทั่วไปของเกมที่มีในปัจจุบันนั้น จะเป็นลักษณะของลูปโปรแกรมซึ่งแท้ที่จริงแล้วเกมก็คือโปรแกรมที่จัดการกับเหตุการณ์ต่างๆ ที่เกิดขึ้นตามที่กำหนดไว้นั่นเอง

เกมที่ผู้จัดทำได้นำมาอินเตอร์เฟสในโครงงานนี้คือเกม Space Shooter ซึ่งเป็นเกมที่ไม่ต้องใช้ทักษะมากมายแต่เป็นเกมที่ให้ความสนุกสนานเพลิดเพลิน มีตัวละครหลักก็คือยานอวกาศที่ใช้สำหรับยิงยานอวกาศฝ่ายตรงข้ามเพื่อให้ได้คะแนน เป็นเกมที่เล่นง่ายๆไม่มีกติกาที่ซับซ้อนมากนักที่เลือกเกมนี้เนื่องจากการเคลื่อนที่ของยานอวกาศที่เป็นตัวหลักสามารถเคลื่อนที่ได้ทุกทิศทางซึ่งตรงกับการตรวจจับการเคลื่อนที่ของสีที่สามารถติดตามสีนั้นไปได้ทุกทิศทางเช่นกัน

3.2.1 ออกแบบโมเดลที่ใช้ในเกม

โมเดลที่ใช้ในเกมนี้แบ่งเป็นสามส่วนดังนี้[9]

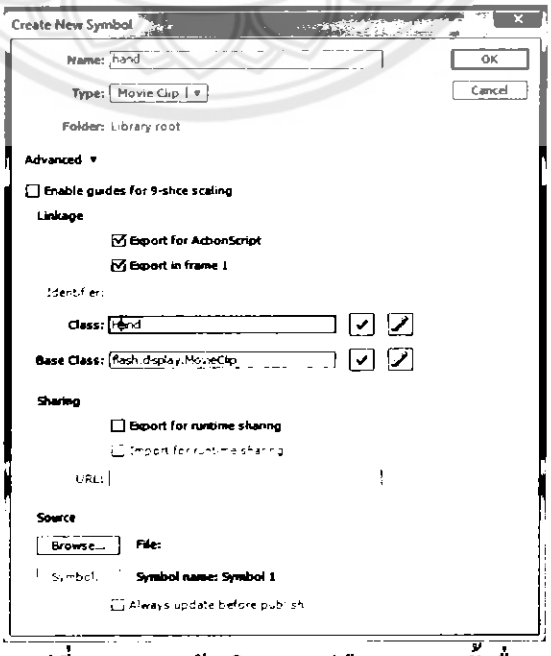
1. โมเดลหลักที่ใช้ในการอินเตอร์เฟส
2. โมเดลที่เป็นฝ่ายตรงข้าม
3. โมเดลที่ใช้เป็นฉาก
4. ส่วนประกอบอื่นๆ

ซึ่งโมเดลเหล่านี้สร้างขึ้นโดยโปรแกรม Flash CS5

1. โมเดลหลักที่ใช้ในการอินเตอร์เฟส มีสองโมเดลคือโมเดลที่เป็นรูปมือที่ใช้สำหรับคลิกเมนูและโมเดลที่เป็นยานอวกาศ ใช้สำหรับยิงยานอวกาศของฝ่ายตรงข้าม

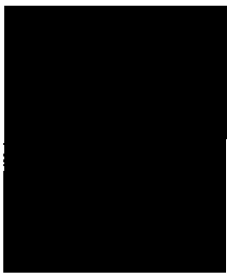
1.1 การสร้างโมเดลมือ

สำหรับการสร้างโมเดลมือมีวิธีการที่ไม่ซับซ้อนมากนัก เริ่มต้นโดยสร้าง Symbol ใหม่โดยคลิกที่เมนู Insert -> New Symbol จะมีหน้าต่างเล็กๆขึ้นมาจากนั้นตั้งชื่อว่า hand เลือกชนิดเป็นแบบ Movie Clip และคลิกเลือกที่ Export for ActionScript แล้วตั้งชื่อคลาสว่า Hand



รูปที่ 3.1 การสร้างโมเดลรูปมือและการตั้งชื่อ

จากนั้นใช้เครื่องมือในการวาดภาพของโปรแกรม Flash วาดภาพให้เป็นรูปมือดังรูปที่ 3.2



รูปที่ 3.2 โมเดลรูปมือหลังจากใช้เครื่องมือในการวาด

จากรูปที่ 3.2 จะเห็นว่าเมื่อเราสร้างโมเดลขึ้นมาใหม่โมเดลนั้นจะถูกเก็บไว้ที่ LIBRARY โดยอัตโนมัติสามารถคลิกลากมาใช้ใน stage หรือเขียนโค้ดเรียกใช้ได้ทันที

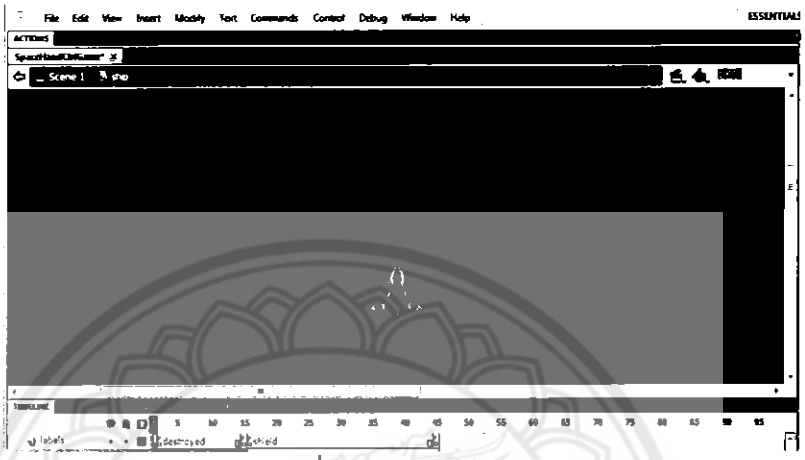
1.2 การสร้างโมเดลยานอวกาศ

สำหรับการสร้างโมเดลยานอวกาศ ค่อนข้างจะมีวิธีการสร้างที่ซับซ้อน มีการกำหนดลาเบล และแอนิเมชั่นเพื่อเพิ่มแอ็คชั่นในเกมและเพิ่มความสมจริงมากขึ้น

โดยขั้นตอนแรกคือวาดตัวละครที่เป็นยานอวกาศขึ้นมา โดยตั้งชื่อ Symbol นี้ว่า Ship เป็นชนิด Movie Clip และ Export for ActionScript ตั้งชื่อคลาสว่า Ship

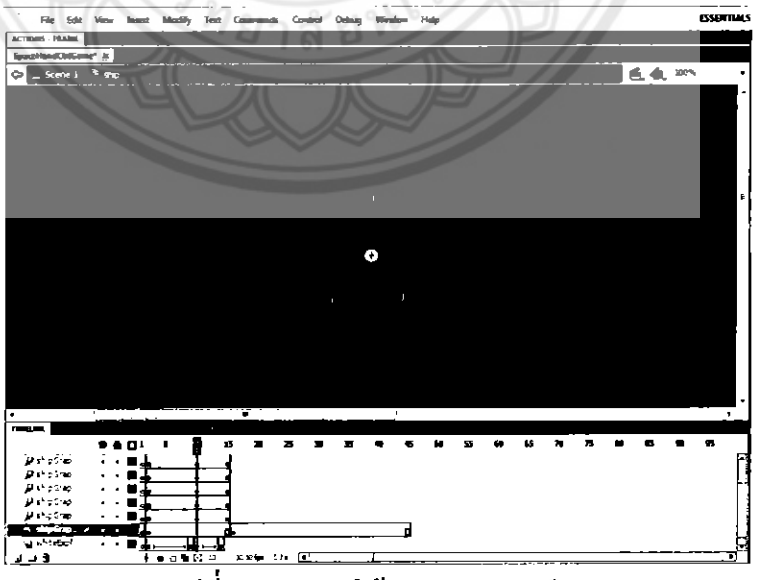
รูปที่ 3.3 การสร้างซิมโบลยานอวกาศ

ต่อมาเป็นการสร้าง Hit Area คือการสร้างพื้นที่สำหรับตรวจสอบการชนกัน โดยจะต้องสร้างซิมโบล hitBox ที่เป็นชนิด Movie Clip ขึ้นมาก่อนจากนั้นลากมาไว้บนสเตจ ในเฟรมที่ 1 เลเยอร์ hitArea จากนั้นจัดให้ครอบคลุมพื้นที่ยานอวกาศแล้วกำหนด Instance Name เป็น hit ดังรูปที่ 3.4



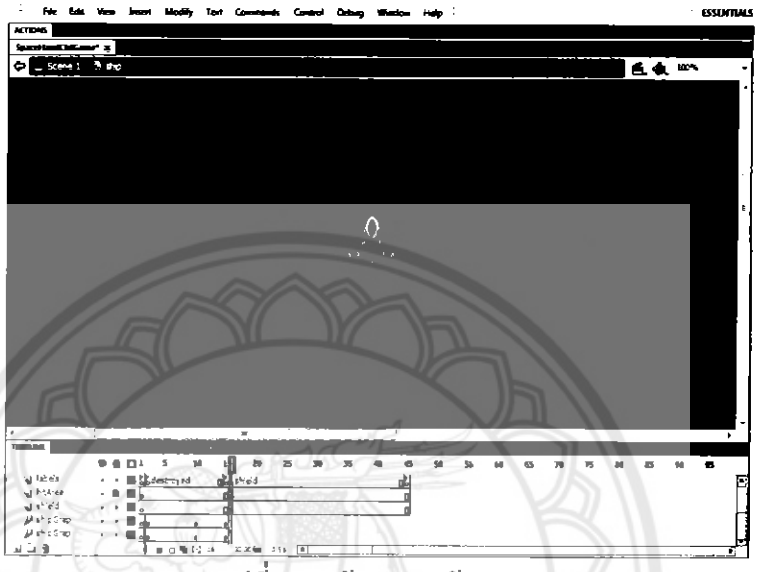
รูปที่ 3.4 สร้าง hit area

จากนั้นทำ motion tween กำหนดให้ยานอวกาศมีการระเบิดเมื่อถูกระเบิดของฝ่ายตรงข้ามหรือเกิดการชนกันโดยการเพิ่มเลเยอร์ขึ้นมา 7 เลเยอร์ แล้วกำหนดให้แต่ละคีย์เฟรมเป็นเสมือนว่ายานกำลังระเบิดดังรูปที่ 3.5



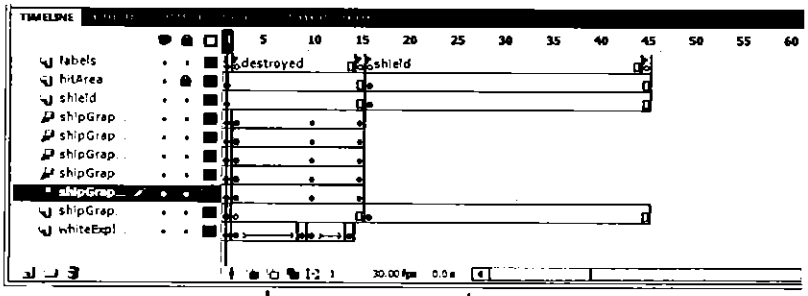
รูปที่ 3.5 การทำให้ยานอวกาศระเบิด

จากรูปที่ 3.5 เป็นการสร้างกราฟิกให้ยานอวกาศมีการระเบิดโดยที่เลเยอร์ shipGraphic1 – shipGraphic5 เป็นการกำหนดทิศทางของชิ้นส่วนยานอวกาศที่ระเบิด เลเยอร์ whiteExplosion เป็นการสร้างให้เกิดแสงคล้ายระเบิดโดยการกำหนด motion tween ส่วนเลเยอร์ shipGraphicจะเป็นการกำหนดให้มีเกราะป้องกันดังรูปที่ 3.6



รูปที่ 3.6 สร้างเกราะป้องกัน

ส่วนการที่จะทำให้เกิดแอ็คชั่นต่างๆจะต้องกำหนดชื่อลาเบลของแต่ละเฟรมที่เกิดการเปลี่ยนแปลงเอาไว้และเรียกใช้ให้เกิดแอ็คชั่นโดยการเขียนโค้ดและระบุชื่อลาเบลเมื่อเกิดการกระทำต่างๆขึ้น การกำหนดชื่อลาเบลจะต้องกำหนดไว้ในเลเยอร์ที่ 1 วิธีการกำหนดคือคลิกที่เฟรมที่ต้องการกำหนดจากนั้นคลิกที่แถบ PROPERTY แล้วกำหนดชื่อลงไป โดยจะกำหนดดังต่อไปนี้คือ เฟรมที่ 1 “default”, เฟรมที่ 2 – 14 “destroyed” , เฟรมที่ 15 destroyedComplete, เฟรมที่ 16 – 45 “shield” ดังรูปที่ 3.7



รูปที่ 3.7 การกำหนดชื่อ label ลงบนเฟรม

2. โมเดลที่เป็นฝ่ายตรงข้าม

ยานอวกาศขนาดเล็กที่สามารถปล่อยระเบิดได้เมื่ออยู่ในระดับเดียวกับยานอวกาศที่เป็นตัวละครหลัก โดยตั้งชื่อว่า stinger และชื่อคลาสคือ Stinger ซึ่งมีวิธีการสร้างเช่นเดียวกันกับยานอวกาศที่เป็นตัวละครหลัก เมื่อสร้างเสร็จแล้วจะได้ยานอวกาศดังรูปที่ 3.8

รูปที่ 3.8 รูปร่างโมเดลยานอวกาศฝ่ายตรงข้าม

3. โมเดลที่เป็นฉากประกอบไปด้วยฉากที่อยู่นิ่งและฉากที่เคลื่อนที่

สำหรับฉากที่อยู่นิ่งเป็นการใช้เครื่องมือวาดภาพธรรมดาจะวาดบนสเตจ หรือสร้างซิมโบลขึ้นมาก่อนแล้วนำมาใช้ก็ได้ สำหรับฉากหลักของเกมเป็นการสร้างซิมโบลก่อนแล้วนำมาวางที่สเตจโดยฉากหลักจะมีลักษณะดังรูปที่ 3.9

รูปที่ 3.9 ฉากหลักของเกม

ส่วนฉากที่มีการเคลื่อนไหวในโครงงานนี้เป็นการจำลองพื้นที่ให้เหมือนกับอวกาศโดยสร้างซิมโบลไว้แล้วเรียกใช้โดยการเขียนโค้ด

ส่วนเมนูต่างๆและการแสดงคะแนนก็เป็นการเรียกใช้โดยการเขียนโค้ดเช่นกันซึ่งเมื่อกำหนดการแสดงผลเรียบร้อยแล้วฉากทั้งหมดจะเป็นดังรูปที่ 3.10

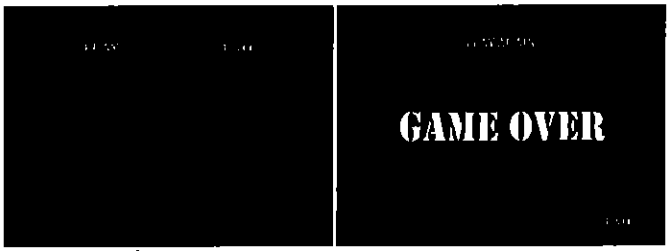
รูปที่ 3.10 ภาพโดยรวมของเกม

4. ส่วนประกอบอื่นๆ

นอกจากตัวละครหลักที่กล่าวมาข้างต้นแล้วภายในเกมยังมีส่วนประกอบ**อื่นๆอีกเช่น กระสุน**ของยานอวกาศตัวหลัก, ระเบิดของฝ่ายตรงข้าม, ตารางแสดงคะแนน, **ฉากเริ่มต้น, ฉากจบ, และเมนูต่างๆ** ดังนี้

รูปที่ 3.11 อาวุธของยานอวกาศตัวหลักและฝ่ายตรงข้าม

รูปที่ 3.12 ตารางแสดงคะแนน

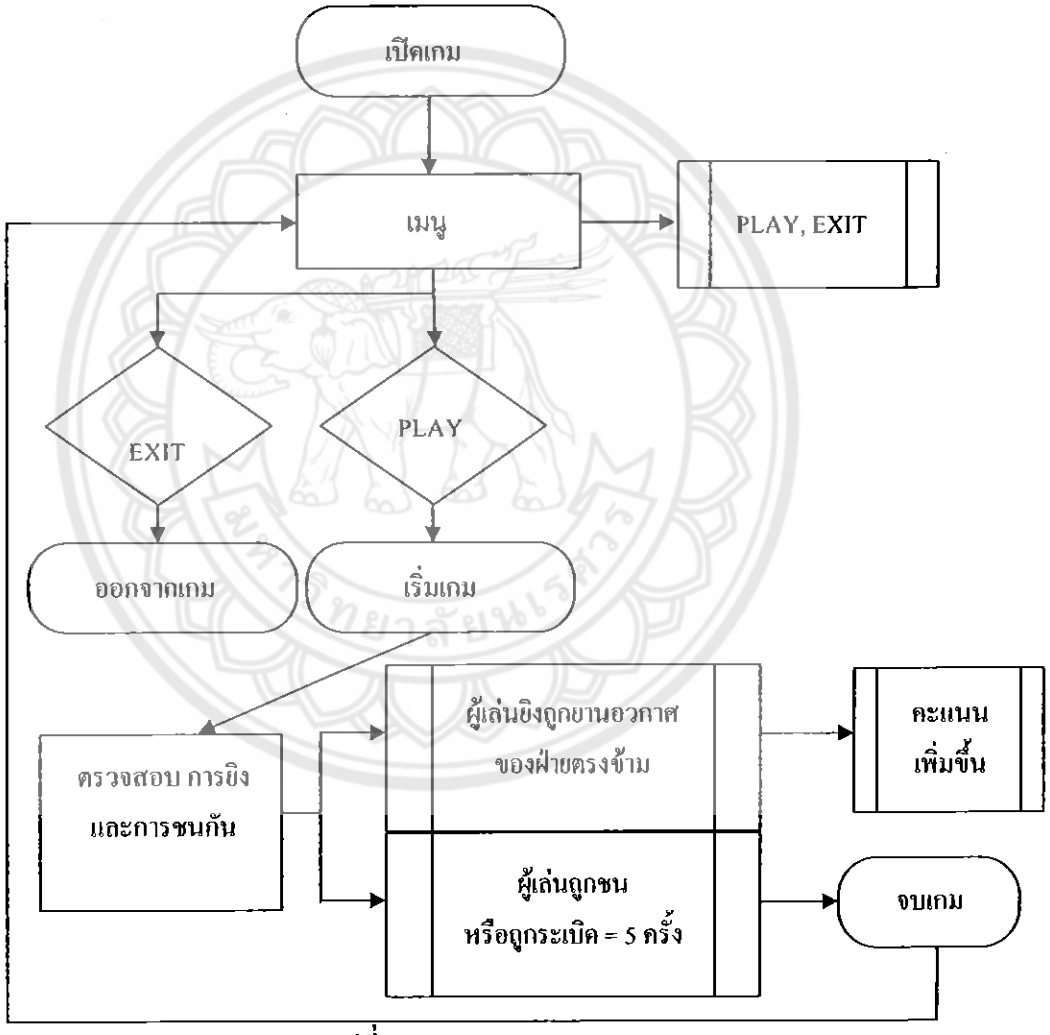


รูปที่ 3.13 ฉากเริ่มต้น ฉากจบและเมนูต่างๆ

3.2.2 การออกแบบเงื่อนไขต่างๆของเกม

ขั้นตอนนี้จะเป็นขั้นตอนที่สำคัญเพราะความสนุกสนานของเกมจะอยู่ตรงที่เงื่อนไขต่างๆ ในการเล่นเกมซึ่งเกม Space Shooter ได้มีการออกแบบรูปแบบหรือเงื่อนไขในการเล่นดังนี้

เกม Space Shooter เป็นเกมที่ให้ผู้เล่นยิงยานอวกาศที่พุ่งลงมา และต้องคอยหลบระเบิดที่ปล่อยออกมาจากยานอวกาศฝ่ายตรงข้ามด้วย ถ้ายิงถูกจะได้คะแนนเพิ่มแต่ถ้าโดนระเบิดหรือชนกับยานอวกาศฝ่ายตรงข้ามถึง5 ครั้งถือว่าจบเกม การออกแบบเกมสามารถอธิบายด้วย Flow Chart ดังรูปที่ 3.14



รูปที่ 3.14 Flow Chart ของเกม

3.2.3 เขียนโปรแกรม

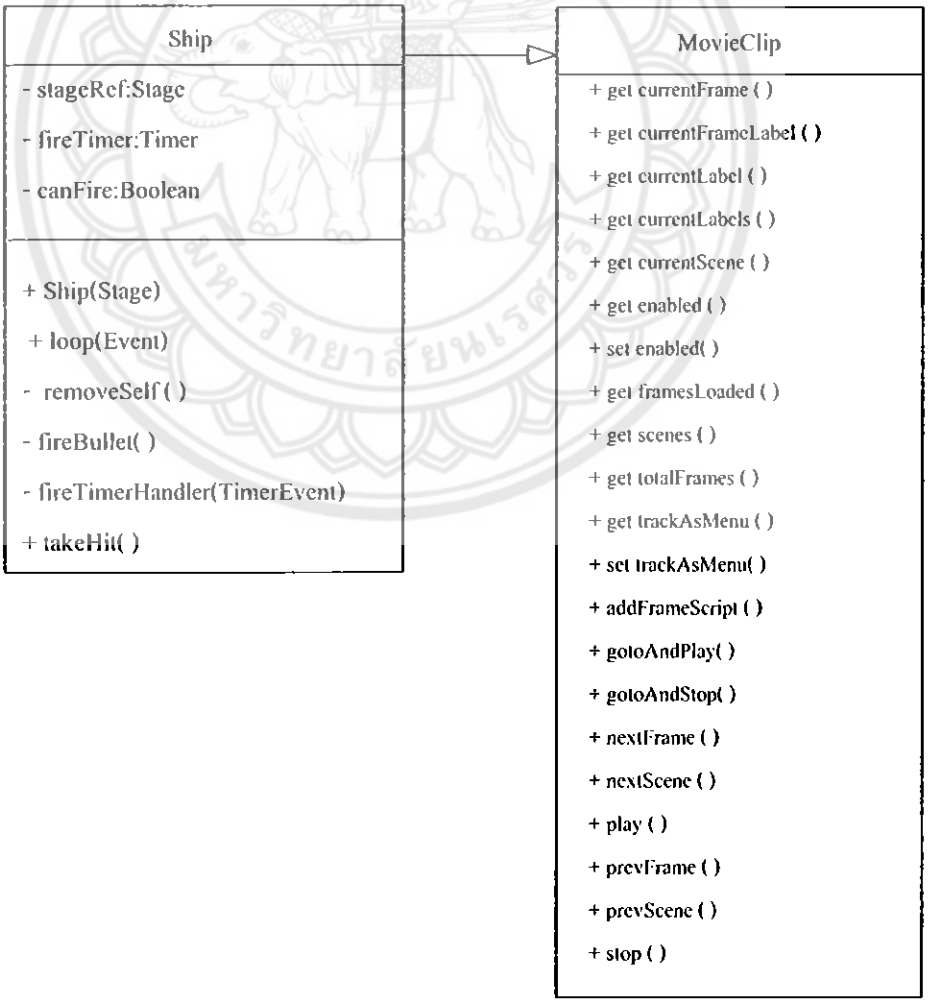
ขั้นตอนนี้เป็นส่วนที่สำคัญที่สุดในการสร้างเกม เพราะต้องมีการเขียนโปรแกรมเพื่อให้เป็นไปตามเงื่อนไขที่ได้กำหนดไว้ ในส่วนการเขียนโปรแกรมแบ่งเป็นขั้นตอนดังนี้

1. การเขียนโปรแกรมเพื่อควบคุมการกระทำของโมเดล

ในขั้นตอนนี้เป็นการเขียนโปรแกรมเพื่อกำหนดแอ็คชั่นให้โมเดล เช่น การกำหนดให้ยานอวกาศระเบิดเมื่อถูกลูกระเบิด, การยิงระเบิดของยานอวกาศ ซึ่งในเกมนี้ยานอวกาศตัวหลักจะยิงเลเซอร์โดยอัตโนมัติ และการปล่อยลูกระเบิดของฝ่ายตรงข้ามจะปล่อยเมื่ออยู่ในแนวเดียวกันกับยานอวกาศตัวหลัก ซึ่งเป็นการเขียนโปรแกรมในคลาสของแต่ละโมเดลโดยตรงและจะไม่ใช้การเขียนโปรแกรมควบคุมโดยเป็นพิมพ์หรือเมาส์

คลาสสำคัญที่ใช้ควบคุมการกระทำของตัวละครได้แก่ คลาส Ship ควบคุมการกระทำของยานอวกาศของผู้เล่น, คลาส Stinger ควบคุมการกระทำของยานอวกาศฝ่ายตรงข้าม, คลาส LaserBlue ควบคุมการยิงเลเซอร์ของยานอวกาศฝ่ายผู้เล่น และคลาส StingerBullet ควบคุมการกระทำของการยิงระเบิดของยานอวกาศฝ่ายตรงข้าม ซึ่งมีรายละเอียดดังต่อไปนี้

a. คลาสไดอะแกรมของยานอวกาศของผู้เล่น



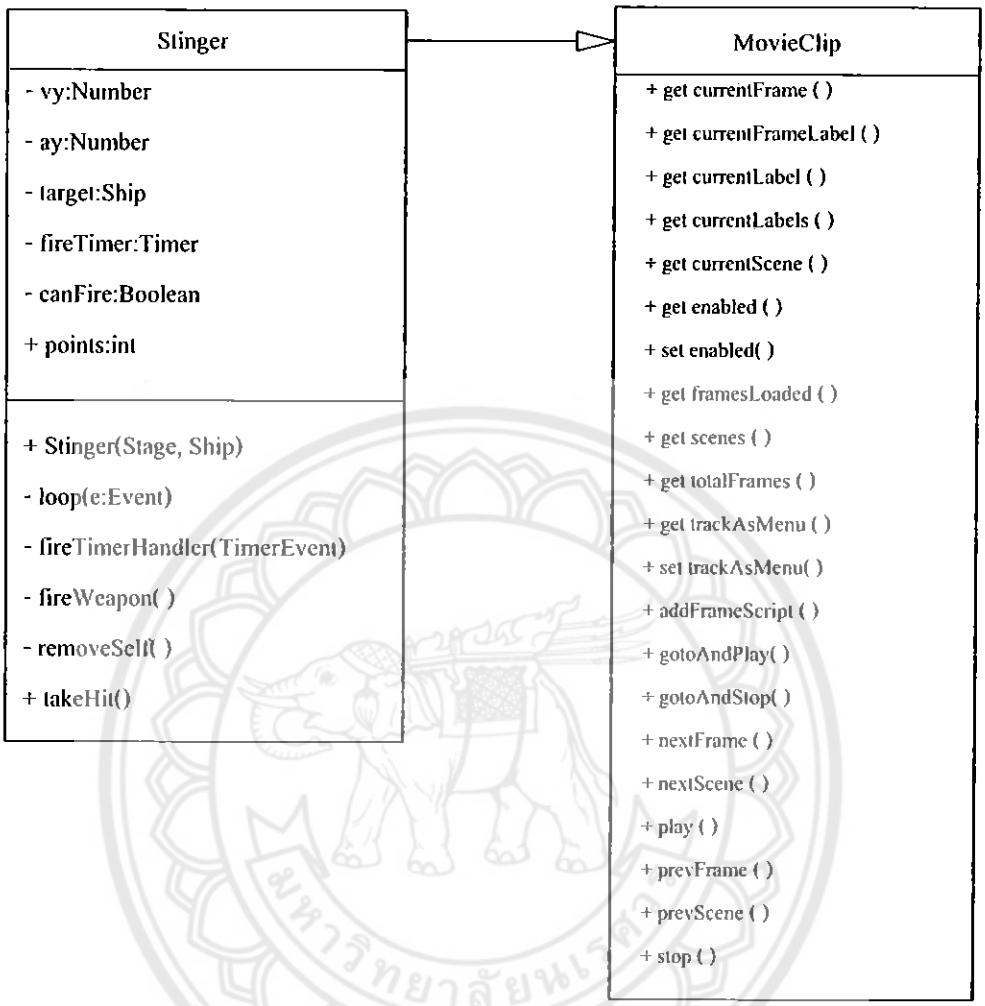
รูปที่ 3.15 คลาสไดอะแกรมของยานอวกาศฝ่ายผู้เล่น

จากคลาสไดอะแกรมอธิบายได้ดังตารางต่อไปนี้

ตารางที่ 3.1 อธิบายส่วนต่างๆในคลาสไดอะแกรม Ship

Data/Method	ความหมาย/หน้าที่
stageRef:Stage	การอ้างถึง stage
fireTimer:Timer	เวลาหน่วงในการยิงเลเซอร์ของยานอวกาศ
canFire:Boolean	การกำหนดการยิง มีได้ 2 ค่าคือ true และ false
Ship(Stage)	เป็นเมธอดที่กำหนดค่าเวลาหน่วงในการยิงเลเซอร์และและกำหนดให้เข้าสู่เมธอด loop เมื่อมีการอ้างอิงถึง
loop(Event)	เป็นการกำหนดการยิงเลเซอร์โดยอัตโนมัติและกำหนดว่าถ้าถูกระเบิดจะเข้าสู่เมธอด removeSelf()
removeSelf( )	เป็นเมธอดสำหรับลบยานอวกาศเมื่อถูกระเบิดหรือถูกชน
fireBullet( )	มีการกำหนดว่าถ้า canFire เป็น true ยานอวกาศสามารถยิงเลเซอร์ได้ ถ้าเป็น false ไม่สามารถยิงได้
fireTimerHandler(TimerEvent)	ถ้าหากเข้าสู่เมธอดนี้จะเป็นการกำหนดให้ยานอวกาศสามารถยิงเลเซอร์ได้
takeHit( )	เป็นการกำหนดให้ยานอวกาศเกิดแอ็คชั่นต่างๆเมื่อถูกระเบิดหรือถูกชน

b. คลาสไดอะแกรมของยานอวกาศฝ่ายตรงข้าม



รูปที่ 3.16 คลาสไดอะแกรมของยานอวกาศฝ่ายตรงข้าม

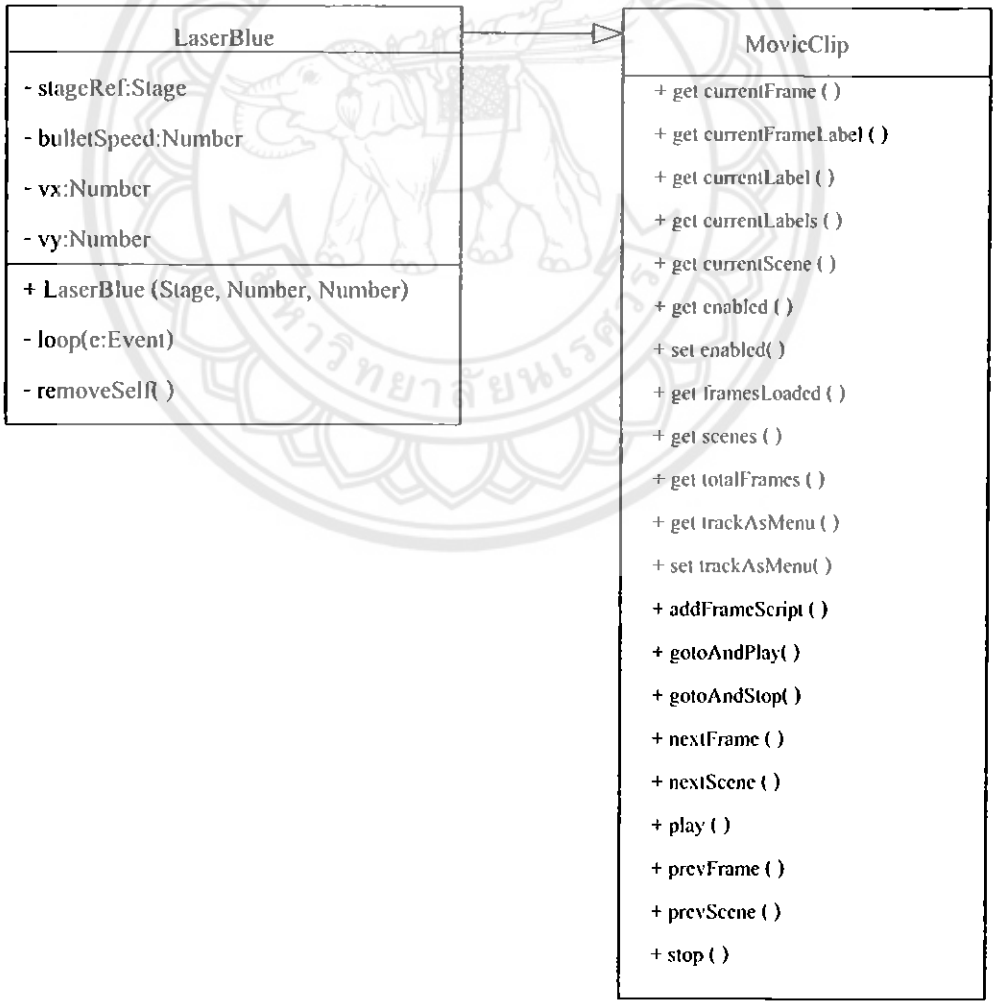
จากคลาสไดอะแกรมอธิบายได้ดังตารางต่อไปนี้

ตารางที่ 3.2 อธิบายส่วนต่างๆในคลาสไดอะแกรม Stinger

Data/Method	ความหมาย/หน้าที่
vy:Number	ความเร็วในแกน y
ay:Number	การเร่งความเร็วในแกน y
target:Ship	กำหนดให้ target เป็น Ship
fireTimer:Timer	เวลาหน่วงในการยิงระเบิดของยานอวกาศ
canFire:Boolean	การกำหนดการยิง มีได้ 2 ค่าคือ true และ false
points:int	กำหนดคะแนนเมื่อยานอวกาศผู้เล่นยิงถูก

Stinger(Stage, Ship)	เป็นการกำหนดตำแหน่ง x, y ที่ยานอวกาศจะเคลื่อนที่ไป
loop(e:Event)	เป็นการกำหนดการกระทำต่างๆของยานอวกาศ
fireTimerHandler(TimerEvent)	ถ้าหากเข้าสู่เมธอดนี้จะเป็นการกำหนดให้ยานอวกาศสามารถยิงระเบิดได้
fireWeapon( )	การกำหนดระยะการปล่อยระเบิด
removeSelf( )	การลบยานอวกาศเมื่อถูกยิง
takeHit( )	เป็นการกำหนดให้ยานอวกาศเกิดแอ็คชั่นต่างๆเมื่อถูกระเบิดหรือถูกชน

c. คลาสไดอะแกรมของการยิงเลเซอร์ฝ่ายผู้เล่น

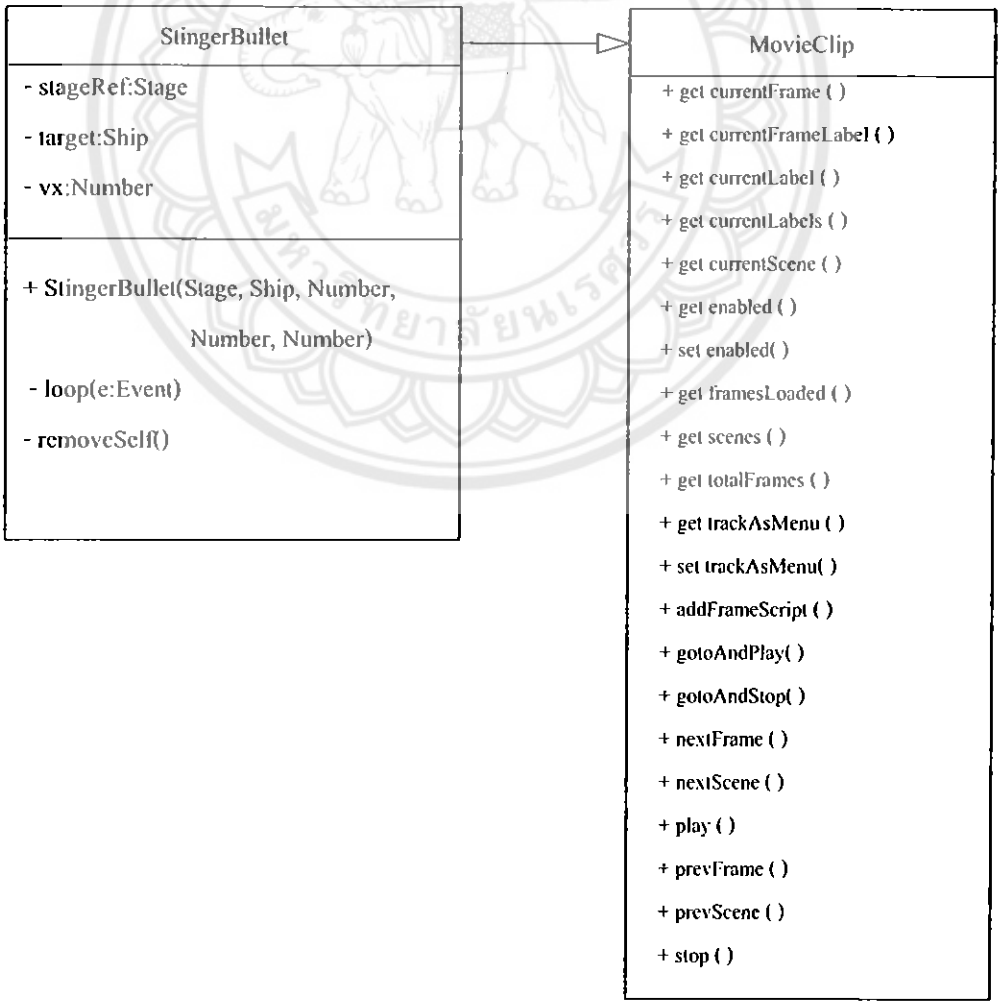


รูปที่ 3.17 คลาสไดอะแกรมของการยิงเลเซอร์ฝ่ายผู้เล่น

ตารางที่ 3.3 อธิบายส่วนต่างๆในคลาสไดอะแกรม LaserBlue

Data/Method	ความหมาย/หน้าที่
stageRef:Stage	การอ้างอิงถึง stage
bulletSpeed:Number	ความเร็วในการยิงเลเซอร์
vx:Number	ความเร็วในการเคลื่อนที่ไปในแกน x
vy:Number	ความเร็วในการเคลื่อนที่ไปแนวแกน y
LaserBlue (Stage, Number, Number)	กำหนดตำแหน่งของการยิงเลเซอร์
loop(e:Event)	เป็นการกำหนดการกระทำต่างๆเมื่อเข้าถึงเมธอด
removeSelf( )	เป็นการหยุดเหตุการณ์ต่างๆใน stage และลบมูฟวี่คลิบออกจาก stage

d. คลาสไดอะแกรมของการยิงระเบิดฝ่ายตรงข้าม



รูปที่ 3.18 คลาสไดอะแกรมของการยิงระเบิดของฝ่ายตรงข้าม

ตารางที่ 3.4 อธิบายส่วนต่างๆในคลาสไดอะแกรม StingerBullet

Data/Method	ความหมาย/หน้าที่
stageRef:Stage	การอ้างอิงถึง stage
target:Ship	เป้าหมายในการอ้างอิงถึงคือคลาส Ship
vx:Number	กำหนดความเร็วในการเคลื่อนที่แนวแกน x
StingerBullet(Stage, Ship, Number, Number, Number)	กำหนดตำแหน่ง x,y และความเร็วในการยิง
loop(Event)	เป็นการกำหนดเหตุการณ์ต่างๆของคลาส
removeSelf ( )	เป็นการหยุดเหตุการณ์ต่างๆใน stage และลบมูฟวี่คลิบออกจาก stage

2. การเขียนโปรแกรมเพื่อควบคุมให้เป็นไปตามเงื่อนไขที่ออกแบบไว้

ขั้นตอนนี้เป็นส่วนของการรวบรวมส่วนประกอบทั้งหมดนำมาใส่เงื่อนไขของเกม และทำให้เกมสามารถดำเนินไปตามรูปแบบที่ได้ออกแบบไว้ได้ ซึ่งในการเขียนโปรแกรมจะเขียนไว้ที่คลาสหลักนั่นก็คือ คลาส ColorTracking ซึ่งจะมีการเขียนฟังก์ชั่นต่างๆเพิ่มเข้าไปดังตารางที่ 3.

ตารางที่ 3.5 ฟังก์ชั่นที่ต้องเขียนเพิ่มเข้าไปในคลาส ColorTracking

ฟังก์ชั่น	หน้าที่
EngineEvent	เป็นการเรียก โมเดลยานอวกาศ, โมเดลมือ, หน้าเมนู และตารางคะแนน เข้ามาแสดงผลในสเตจ
loop	เป็นฟังก์ชั่นหลักในการควบคุมการทำงานของเกม เมื่อเข้ามาในลูปยานอวกาศฝ่ายตรงข้ามจะเริ่มทำงาน และเป็นการกำหนดว่าจะให้มีการจบเกมเมื่อใด ถ้าเงื่อนไขตรงกับในลูป ในที่นี้คือ Lives = 0 จะจบเกมทันที
playGame	เป็นการกำหนดให้เริ่มเกมและทดสอบการชนกันหรือการได้คะแนน
playGame2	เมื่อเกมจบลงครั้งแรก ถ้าผู้เล่นเลือก play อีกครั้งจะเข้าสู่ฟังก์ชั่น playGame2 และเริ่มการวนลูปเกมอีกครั้ง คะแนนจะเริ่มที่ 0 , คะแนนสูงสุดจะเท่ากับคะแนนมากสุดที่ทำได้ล่าสุด และ Lives จะถูกเซตกลับเป็น 5
enemyKilled	เป็นการอัพเดทคะแนนในการยิงถ้ายิงถูกยานอวกาศฝ่ายตรงข้ามคะแนนจะเพิ่มขึ้น

ตารางที่ 3.5 (ต่อ) ฟังก์ชั่นที่ต้องเขียนเพิ่มเข้าไปในคลาส ColorTracking

ฟังก์ชั่น	หน้าที่
removeEnemy	เป็นการลบยานอวกาศที่ถูกยิง
shipHit	เมื่อยานอวกาศของผู้เล่นถูกชน จะถูกอัพเดทไปยังฟังก์ชั่น updateHits ในคลาส ScoreHUD ซึ่ง Lives จะลดลงครั้งละ 1

3.2.4 วิธีการเล่นเกม

1. เริ่มต้นผู้เล่นจะมี Lives เท่ากับ 5 แต่จะลดลงทีละ 1 เมื่อถูกระเบิดหรือชนกับยาน - อวกาศ ของฝ่ายตรงข้าม
2. ยานอวกาศของผู้เล่นจะปล่อยเลเซอร์โดยอัตโนมัติ ถ้าหากผู้เล่นยิงโดนยานอวกาศของ ฝ่ายตรงข้ามจะได้คะแนนเพิ่มขึ้น
3. ถ้า Lives เท่ากับ 0 จบเกม ผู้เล่นสามารถเลือกเล่นเกมใหม่หรือออกจากเกมได้

3.3 ปรับปรุงโปรแกรม ColorTracking ให้เหมาะสมกับเกมที่ออกแบบไว้ และอินเตอร์เฟสกับเกม

ในส่วนนี้ได้มีการนำโปรแกรม ColorTracking ที่ได้ศึกษาไปแล้วข้างต้นมาปรับปรุงให้เหมาะสมกับเกมที่ออกแบบไว้ และนำมาปรับใช้ในการอินเตอร์เฟสกับเกมเพื่อให้ควบคุมการเคลื่อนที่ของตัวละครในเกมได้ซึ่งมีขั้นตอนดังนี้

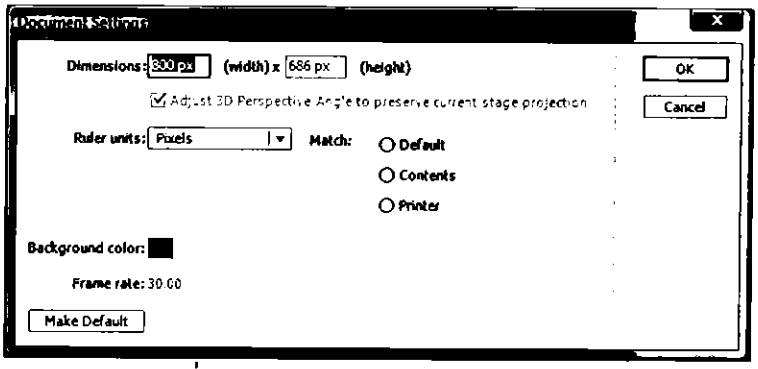
1. ปรับปรุงการแสดงผลบนหน้าจอ
2. เขียนฟังก์ชั่นให้โมเดลที่เตรียมไว้ ติดตามการเคลื่อนที่ของมือ

3.3.1 ปรับปรุงการแสดงผลบนหน้าจอ

ในส่วนนี้เป็นการปรับรูปแบบการแสดงผลให้เข้ากับเกมที่ได้ออกแบบไว้โดยมีการปรับปรุงทั้งหมดดังนี้

3.3.1.1 ปรับขนาดการแสดงผลของหน้าจอ

ในการแสดงผลทางหน้าจอของโปรแกรม ColorTracking มีขนาด 630 พิกเซล x 525 พิกเซล ปรับให้เหมาะสมกับเกมที่ออกแบบไว้เป็น 800 พิกเซล x 686 พิกเซล มีวิธีการปรับดังนี้คือที่เมนูบาร์เลือก Modify -> Document จะมีหน้าต่างเล็กๆขึ้นมาให้ใส่ค่าที่ต้องการลงไป ดังรูปที่ 3.19



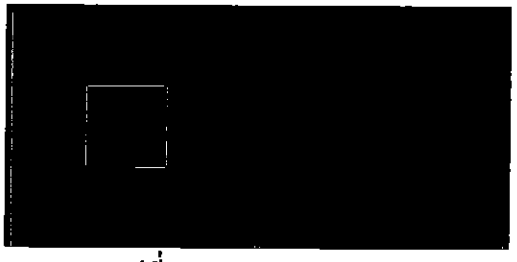
รูปที่ 3.19 การกำหนดขนาดจอแสดงผล

3.3.1.2 ปรับขนาดและตำแหน่งการแสดงผลของกล้องเว็บแคม

ในเกมที่ออกแบบไว้ตัวละครจะต้องเคลื่อนที่ไปได้โดยรอบพื้นที่ ดังนั้นการติดตามการเคลื่อนที่ของมือก็จะต้องติดตามไปได้โดยรอบเช่นกัน ซึ่งโปรแกรม ColorTracking จะมีการแบ่งพื้นที่แสดงผลวีดีโอเป็นสองส่วนคือส่วนที่เป็นภาพวีดีโอที่รับเข้ามาและส่วนที่เป็นภาพบิตแมปที่มีการติดตามสีที่เลือก จึงจำเป็นจะต้องปรับตำแหน่งและขนาดใหม่เพื่อให้เหมาะสมกับขนาดของจอภาพที่กำหนดขึ้นใหม่และให้เหมาะสมกับพื้นที่ในการเล่นเกม

รูปที่ 3.20 การแสดงผลของโปรแกรม ColorTracking

พื้นที่ที่แสดงผลภาพวีดีโอและภาพแบบบิตแมปจะมีการกำหนด Instance Name ไว้คือ block1 และ block 2 ตามลำดับ



รูปที่ 3.21 block1, block2

สำหรับการปรับจะปรับให้เหลือเพียงแค่ block 1 ซึ่งจะแสดงทั้งภาพวีดีโอและภาพบิตแมป แต่เมื่อกำหนดให้แสดงผลพร้อมกันทั้งภาพวีดีโอและภาพบิตแมปภาพวีดีโอจะอยู่ด้านบนของภาพบิตแมปทำให้มองไม่เห็น

รูปที่ 3.22 การปรับให้เหลือเพียง block1

จากรูปที่ 3.22 กรอบสี่เหลี่ยมสีฟ้าคือ block 1 ซึ่งเป็นการอ้างอิงตำแหน่งมุมบนซ้ายของการแสดงผลทั้งภาพวีดีโอและภาพบิตแมป

ส่วนขนาดของภาพวีดีโอและภาพบิตแมปจะมีขนาดเท่ากันโดยปรับที่ขนาดความกว้างและความสูงของวีดีโอในคลาส CameraDisplay

3.3.1.3 นำปุ่มและส่วนที่ใช้ปรับภาพทั้งหมดออก

ในขั้นตอนนี้เป็นการนำส่วนที่ไม่ได้ใช้ในเกมออกคือปุ่มคลิกเลือกสีและส่วนที่ใช้สำหรับปรับสีเนื่องจากมีการปรับโดยการตั้งค่าที่เหมาะสมในการตรวจจับและติดตามการเคลื่อนที่ของมือไว้แล้ว เมื่อมืออยู่ในตำแหน่งที่กล้องเว็บแคมกำลังจับภาพจะมีการตรวจจับสีและติดตามการเคลื่อนที่โดยอัตโนมัติ

รูปที่ 3.23 จอภาพก่อนและหลังการนำส่วนที่ไม่เกี่ยวข้องออก

ในการปรับภาพให้มีความคมชัดขึ้นมีการกำหนดค่า contrast, saturation, blur, tolerance, และ brightness ไว้ดังนี้

```
private function setupControls():void
        {
                contrast = new HSlider(this,0,0,updateFilters);
                contrast.setSliderParams(0,10,5);
                contrast.x = 1500;
                contrast.y = 1500;

                saturation = new HSlider(this,0,0,updateFilters);
                saturation.setSliderParams(0,200,50);
                saturation.x = 1500;
                saturation.y = 1500;

                blur = new HSlider(this,0,0,updateFilters);
                blur.setSliderParams(0,10,10);
                blur.x = 1500;
                blur.y = 1500;

                tolerance = new HSlider(this,0,0,updateFilters);
                tolerance.setSliderParams(0,200,50);
                tolerance.x = 1500;
                tolerance.y = 1500;

                brightness = new HSlider(this,0,0,updateFilters);
                brightness.setSliderParams(0,100,15);
                brightness.x = 1500;
                brightness.y = 1500;

                updateFilters();
        }
```

จากโค้ดเดิมที่เป็นการใช้ HUI ให้สามารถปรับสีของวีดีโอได้ สิ่งที่เป็น HUI คือ HSlider ซึ่งจะเป็นในลักษณะสไลด์ที่ใช้เลื่อนปรับค่าต่างๆ ซึ่งในเกมที่ออกแบบไว้จะต้องตัดส่วนนี้ออกไป

และตั้งค่าไว้ให้เหมาะสม ค่าที่มีผลต่อการค้นหาสีมือคือค่า saturation และค่า tolerance ในการปรับจะต้องทดสอบปรับค่าทั้งสองค่านี้จนได้ค่าที่เหมาะสมในการจับสีของมือ และค่าที่เราตั้งไว้นี้จะอัพเดทไปสู่ภาพวีดีโอและแสดงผลดังโค้ดต่อไปนี้

```
private function updateFilters(e:Event=null):void
{
        cm = new ColorMatrix();
        cm.adjustContrast(contrast.value);
        cm.adjustSaturation(saturation.value);
        cm.adjustBrightness(brightness.value);
        camDisp1.filters = [new ColorMatrixFilter(cm),
        newBlurFilter(blur.value,blur.value,1)];
        ci.tolerance = tolerance.value;
}
```

หลังจากปรับปรุงภาพจากโค้ดที่กล่าวมาข้างต้นภาพวีดีโอมีการเปลี่ยนแปลงดังรูปที่ 3.24

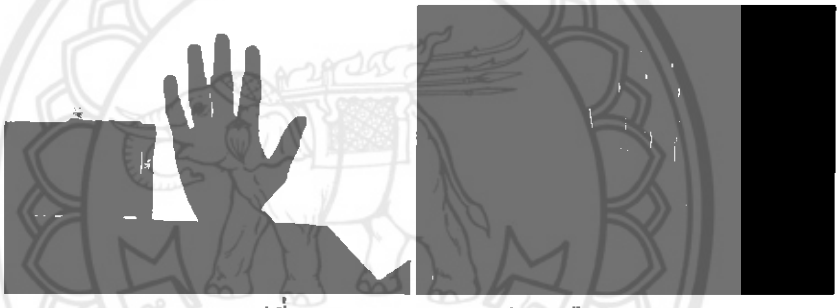
รูปที่ 3.24 รูปภาพวีดีโอก่อนและหลังจากมีการปรับปรุงภาพ

การกำหนดสีที่จะติดตาม(มือ) มีการกำหนดไว้ดังนี้

```
private function setColor(e:MouseEvent = null):void
{
        rect.visible = false;
        hand.visible = false;
        ourShip.visible = true;
        camDisp1.buttonMode = false;
```

```
var bmd:BitmapData = new BitmapData(vWidth,vHeight,true);
bmd.draw(camDisp1);

var c:ColorTransform = new ColorTransform();
c.color = 0xFC7850;//FC = red, 78 = Green, 50 = blue (252,120,80)
ci.color = c.color;
}
```

จากโค้ดเป็นการกำหนดสีที่ต้องการโดยตรงผ่านตัวแปร ci ซึ่งมาจากการเรียกใช้คลาส ColorIsolator ค่าสีนี้จะเป็นค่าสีเป้าหมายที่ถูกติดตามโดยผ่านเมธอด getColorBoundsRect() ซึ่งค่าสีที่กำหนดเป็นค่าสีที่ใกล้เคียงกับสีของมือผลที่ได้แสดงดังรูปที่ 3.25



รูปที่ 3.25 การตรวจจับสีของมือ

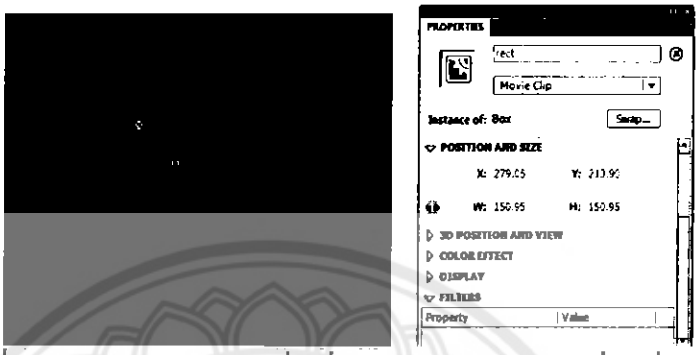
3.3.2 เขียนฟังก์ชั่นให้โมเดลที่เตรียมไว้ติดตามการเคลื่อนที่ของมือ

ในโปรแกรม ColorTracking มีฟังชั่นที่ใช้กรอบสี่เหลี่ยมในการติดตามสีซึ่งเขียนไว้ที่คลาส ColorTracking ที่เป็นคลาสหลักโดยการสร้างซิมโบลกรอบสี่เหลี่ยมแบบ MovieClip มาวางบนสเตจหลักแล้วตั้งชื่อ Instance ว่า rect มีรูปแบบการเรียกใช้ดังต่อไปนี้

```
private function handleEnterFrame3(e:Event):void
{
        // process the bitmap data and display it
        bmDisp.update(ci.process(camDisp.bitmapData));
        // tween the rectangle
        TweenLite.to(rect,0.15, {x:bmDisp.x+ci.rect.x,
                                 y:bmDisp.y+ci.rect.y,
                                 width:ci.rect.width,
                                 height:ci.rect.height});
```

/*กรอบสี่เหลี่ยมจะเคลื่อนที่ไปตามตำแหน่ง x เท่ากับตำแหน่งของภาพบิตแมปที่จะติดตามและขนาดของสี่เหลี่ยมจะเท่ากับขนาดของพื้นที่สีที่ได้กำหนดให้ติดตามเช่นกัน และมีค่าหน่วงเวลาในการเคลื่อนที่เท่ากับ 0.15 วินาที */

}



รูปที่ 3.26 การกำหนดกรอบสี่เหลี่ยมในการติดตามการเคลื่อนที่ของสี

จากการใช้ เมธอด getColorBoundRect จะมีกรอบสี่เหลี่ยมล้อมรอบมือและฟังก์ชั่น TweenLite จะเป็นการติดตามการเคลื่อนที่ของมือ ดังรูปที่ 3.27

รูปที่ 3.27 แสดงผลการใช้กรอบสี่เหลี่ยมติดตามการเคลื่อนที่ของมือ

3.3.2.1 เขียนฟังก์ชั่นของโมเดลมือ

```
// process the bitmap data and display it
bmDisp1.update(ci.process(camDisp1.bitmapData));
// tween the rectangle
TweenLite.to(hand,0.15, {x:bmDisp1.x+ci.rect.x,
y:bmDisp1.y+ci.rect.y,
```

```
width:hand.width,
height:hand.height});
```

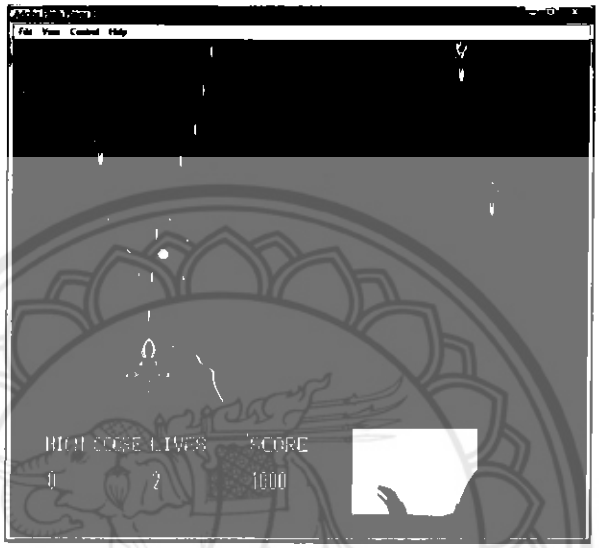
จากโค้ดเป็นบางส่วนของฟังก์ชั่น HandleEnterFrame เป็นการใช้โมเดลมือติดตามการเคลื่อนที่ของสีซึ่งมีรูปแบบการเขียนคล้ายกับฟังก์ชั่น HandleEnterFrame3 เพียงแต่เป็นการนำโมเดลมือมาติดตามแทนและกำหนดให้ขนาดของโมเดลขณะติดตามสี เท่ากับขนาดของโมเดลจริงที่สร้างขึ้นซึ่งไม่มีการเปลี่ยนแปลงขนาดเหมือนกับฟังก์ชั่น HandleEnterFrame3 แสดงผลดังรูปที่ 3.28

รูปที่ 3.28 การใช้โมเดล hand ในการติดตามการเคลื่อนที่ของมือ

3.3.2.2 เขียนฟังก์ชั่นของโมเดลยานอวกาศ

```
private function handleEnterFrame2(e:Event):void
{
        bmDisp1.update(ci.process(camDisp1.bitmapData));
        // tween the rectangle
        TweenLite.to(ourShip,0.15, {x:bmDisp1.x+ci.rect.x,
        y:bmDisp1.y+ci.rect.y,
        width:ourShip.width,
        height:ourShip.height});
}
```

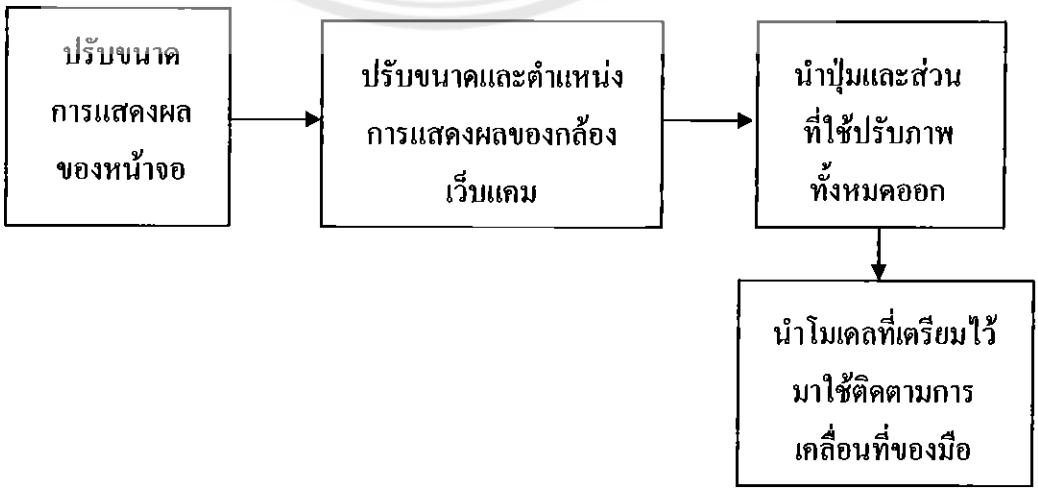
จากโค้ดเป็นบางส่วนของฟังก์ชั่น HandleEnterFrame2 เป็นการใช้โมเดลยานอวกาศ (ourShip) ติดตามการเคลื่อนที่ของสีซึ่งมีรูปแบบการเขียนคล้ายกับฟังก์ชั่น HandleEnterFrame แต่ ourShip จะประกาศเป็นชนิดคลาส ไม่ได้เป็น MovieClip เหมือนกับ rect และ hand เนื่องจากจะต้องมีการเขียนคำสั่งควบคุมการกระทำต่างๆในคลาส Ship เรียกใช้โมเดลนี้โดยการเขียนโค้ดแสดงผลดังรูปที่ 3.29



รูปที่ 3.29 การใช้โมเดล ourShip ในการติดตามการเคลื่อนที่ของมือ

3.3.3 คลาสไดอะแกรมของคลาสหลัก

คลาสไดอะแกรมของคลาสหลักที่จะแสดงต่อไปนี้เป็นคลาสที่รวบรวมการปรับปรุงโปรแกรมเพื่อนำมาใช้ร่วมกับเกมอธิบายได้ด้วย Block Diagram ดังนี้



รูปที่ 3.30 Block Diagram ของการปรับปรุงโปรแกรมเพื่อนำมาใช้กับเกม

3.4 Publish โปรแกรม

เมื่อเขียนโปรแกรมเสร็จทั้งหมดแล้วจะต้อง Publish โปรแกรมเพื่อให้สามารถนำมาเผยแพร่ได้โดย รูปแบบไฟล์ฟอร์แมตในงานนี้จะต้อง Publish เป็นไฟล์ .swf ซึ่งมีวิธีการดังนี้

ที่เมนูบาร์ เลือก File -> Publish หรือกด Alt+Shift+F12 หรือ กด Ctrl + Enter เมื่อพับลิชเรียบร้อยแล้วจะได้ไฟล์ .swf ดังรูปที่ 3.31 และสามารถนำไปใช้งานได้ทันที

SpaceHandCtrlGame
SWF Movie

รูปที่ 3.31 ไฟล์ .swf ที่ได้จากการพับลิชโปรแกรม

3.5 ทฤษฎีและหลักการต่างๆที่นำมาใช้ในขั้นตอนการตรวจจับและติดตามสี

ทฤษฎีและหลักการที่สำคัญสำหรับการตรวจจับและติดตามสีสรุปได้ตามตารางดังนี้

ตารางที่ 3.6 สรุปทฤษฎีและหลักการต่างๆที่นำมาใช้ในขั้นตอนการตรวจจับและติดตามสี

ทฤษฎี/หลักการ	การนำมาใช้งาน
Color Enhancement	ใช้ในการปรับปรุงรูปภาพให้มีความคมชัดตามต้องการโดยใช้ไลบารี่ bit 101 สร้าง UI ในการปรับค่าต่างๆ
Bitmap Data	ใช้สำหรับการประมวลผลภาพบิตแมปในระดับระดับพิกเซล
Color Matrix	ใช้ในการจัดการสีจาก Bitmap Data
Color Isolation,	ใช้สำหรับแยกสีที่ต้องการ
Threshold	เป็นเมธอดหนึ่งในคลาส BitmapData ใช้ในขั้นตอนการทำ Color Isolation
getColorBoundsRect	เป็นเมธอดหนึ่งในคลาส BitmapData ใช้สำหรับกำหนดขอบเขตของสีที่ระบุ
TweenLite	เป็นไลบรารีที่นำมาใช้ในการติดตามการเคลื่อนที่ของมือที่ได้จากการกำหนดขอบเขตโดยเมธอด getColorBoundsRect

บทที่ 4
ผลการทดสอบโปรแกรม

ในบทนี้จะกล่าวถึงรายละเอียดต่างๆของการทดสอบความแม่นยำของการประมวลผลว่ามีข้อผิดพลาดในเรื่องใดบ้าง และการทดสอบเกมที่สร้างขึ้น ว่าเป็นไปตามเงื่อนไขที่ออกแบบไว้หรือไม่เพื่อที่จะได้หาทางแก้ไขปรับปรุงต่อไป

สำหรับการทดสอบแบ่งเป็นสองส่วนคือ ส่วนของการประมวลผลภาพ และส่วนการทำงานของเกม การทดสอบการประมวลผลภาพ เป็นการทดสอบเล่นเกมในสถานที่และตำแหน่งในการติดตั้งกล้องเว็บแคมแตกต่างกันไป เพื่อวิเคราะห์ผลของการตรวจจับและติดตามการเคลื่อนที่ของมือ เนื่องจากภาพที่รับเข้ามาใช้ในการประมวลผลเป็นภาพวีดีโอ ซึ่งจะมีการเปลี่ยนแปลงคุณภาพไปตามสภาพแวดล้อม ในส่วนการทดสอบการทำงานของเกมจะทดสอบว่าเกมที่สร้างขึ้นเป็นไปตามเงื่อนไขต่างๆที่ได้ออกแบบไว้หรือไม่โดยการทดสอบในสถานที่ๆมีสภาพแวดล้อมที่เหมาะสมและตำแหน่งในการติดตั้งกล้องเว็บแคมที่เหมาะสม ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 ผลการทดสอบความแม่นยำในการประมวลผล

ในการทดสอบความแม่นยำในการประมวลผลได้ทดสอบในสถานที่และวิธีการแตกต่างกันไปคือ ทดลองเล่นเกมในห้องที่มีความสว่างมาก, ทดลองเล่นเกมในห้องที่มีความสว่างน้อย, ทดลองเล่นเกมในสถานที่ที่ด้านหลังมีวัตถุอื่นๆปะปน และทดลองเล่นเกมโดยติดตั้งกล้องเว็บแคมไว้ในระดับที่มองเห็นใบหน้าซึ่งมีผลการทดสอบดังนี้

1. จากการทดสอบการเล่นเกมในห้องที่มีความสว่างมากหรือกล้องอยู่ในตำแหน่งย้อนแสงทำให้ภาพที่รับมามีความสว่างมากเกินไปสีของมือจึงไม่ตรงกับสีที่ได้ระบุไว้ให้มีการติดตาม จึงไม่สามารถติดตามการเคลื่อนที่ของมือได้

2. จากการทดลองเล่นเกมในห้องที่มีความสว่างน้อยเกินไป ภาพที่ได้รับจากกล้องวีดีโอมีความมืดมากเกินไปทำให้สีของมือมีสีเข้มขึ้น แต่ยังมีบางส่วนของมือที่มีสีตรงกับสีที่ระบุไว้ จึงทำให้โมเดลที่ติดตามการเคลื่อนที่ของมือมีการขยับได้บ้างแต่ควบคุมได้ไม่ดีนัก ทำให้ไม่สามารถเล่นเกมได้

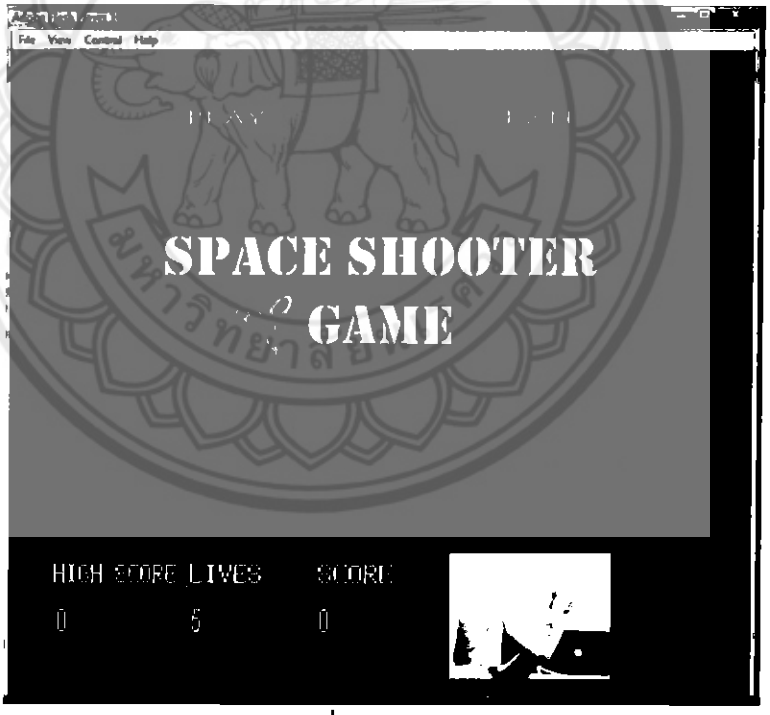
3.จากการทดลองเล่นเกมในสถานที่ที่ด้านหลังมีวัตถุอื่นๆปะปน ถ้าหากด้านหลังมีโทนสีใกล้เคียงกับสีของมือ เช่น แดง, ส้ม, เหลือง, น้ำตาลอ่อน จะทำให้เกิดความคลาดเคลื่อนในการติดตามและไม่สามารถควบคุมตัวละครได้ เนื่องจากโปรแกรมจะติดตามสีที่มีความใกล้เคียงกับสีของมือที่กำหนดไว้ทั้งหมด

4. จากการทดสอบเล่นเกมโดยติดตั้งกล้องเว็บแคมไว้ในระดับที่มองเห็นใบหน้า ในห้องที่มีสภาพแสงที่เหมาะสม พบว่าสามารถตรวจจับสีและการเคลื่อนที่ได้แต่เป็นการตรวจจับทั้งใบหน้าและมือรวมกันการควบคุมตัวละครในเกมโดยใช้มือจึงเกิดความผิดพลาด

4.1.1 รูปภาพแสดงการประมวลผลภาพที่เกิดจากการทดลองเล่นเกมในสถานที่และวิธีการที่แตกต่างกันไป

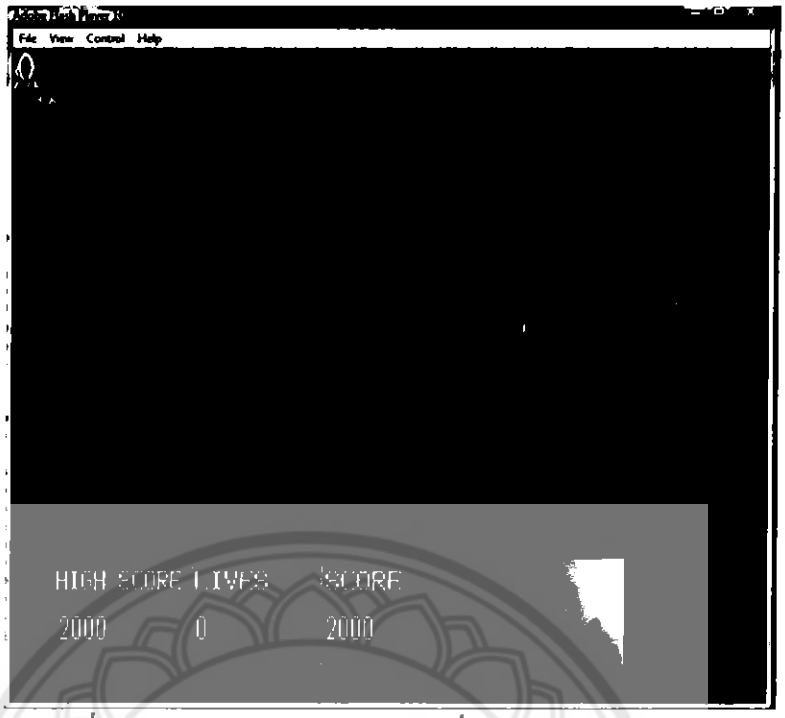
ในการทดลองดังที่ได้กล่าวมาแล้วในหัวข้อที่ 4.1 เป็นการทดลองโดยการใช้กรอบสี่เหลี่ยมให้ตรวจจับและติดตามการเคลื่อนที่ของมือ ว่ามีความแม่นยำมากแค่ไหนและสภาพแวดล้อมที่ใช้ในการเล่นเกมควรเป็นอย่างไรถึงจะเหมาะสม ได้ผลการทดลองดังต่อไปนี้

1.การทดสอบการเล่นเกมในห้องที่มีความสว่างมากหรือกล้องอยู่ในตำแหน่งย้อนแสง จะเห็นว่าภาพวีดีโอมีความสว่างมากทำให้สีของมือไม่ตรงกับสีที่ระบุไว้



รูปที่ 4.1 การทดสอบการเล่นเกมในห้องที่มีความสว่างมากหรือกล้องอยู่ในตำแหน่งย้อนแสง

2. การทดลองเล่นเกมในห้องที่มีความสว่างน้อยเกินไป จะเห็นว่าภาพที่รับเข้ามามืดเกินไป แต่ยังสามารถขยับตัวละครได้แต่ไม่สามารถควบคุมทิศทางการเคลื่อนที่ได้



รูปที่ 4.2 การทดลองเล่นเกมในห้องที่มีความสว่างน้อยเกินไป

3. การทดลองเล่นเกมในสถานที่ที่ด้านหลังมีวัตถุอื่นๆปะปน จะเห็นว่ากรอบสี่เหลี่ยมตรวจจับสีที่ใกล้เคียงกับมือทำให้การเคลื่อนที่ของตัวละครผิดพลาด

รูปที่ 4.3 การทดสอบเล่นเกมในสถานที่ที่ด้านหลังมีวัตถุอื่นๆปะปน

4. การทดสอบเล่นเกมโดยติดตั้งกล้องเว็บแคมไว้ในระดับที่มองเห็นใบหน้าจะเห็นว่ากรอบสี่เหลี่ยมตรวจจับสีที่ใกล้เคียงกับมือทำให้การเคลื่อนที่ของตัวละครผิดพลาดเช่นเดียวกันกับการทดลองในข้อที่ 3

รูปที่ 4.4 การทดสอบเล่นเกมโดยติดตั้งกล้องเว็บแคมไว้ในระดับที่มองเห็นใบหน้า

4.2 ผลการทดลองขั้นตอนการทำงานของเกม

จากการทดสอบเล่นเกมในสถานที่ที่เหมาะสม ซึ่งสถานที่ในที่นี้คือในห้องที่มีแสงสว่างเพียงพอ ด้านหลังผู้เล่นไม่มีวัตถุอื่นๆที่มีโทนสี ส้ม, แดง, เหลือง, น้ำตาล และติดตั้งกล้องไว้ในระดับที่มองไม่เห็นใบหน้าและไม่อยู่ตำแหน่งที่ย้อนแสง แบ่งเป็นการทดลองดังนี้

1. การควบคุมตัวละครและการทำงานร่วมกับการประมวลผลภาพ จากการปรับปรุงโปรแกรม ColorTracking เพื่อให้สามารถนำมาใช้กับเกมได้นั้น ได้มีการปรับปรุงรูปแบบการแสดงผลทางหน้าจอ และได้นำ UI ต่างๆในโปรแกรมออก ให้เป็นการตรวจจับและติดตามการเคลื่อนที่ของมือโดยอัตโนมัติ และนำส่วนของเกมเข้ามาอินเตอร์เฟสกับโปรแกรม เมื่อคอมพิวเตอร์รับภาพวีดีโอเข้ามา จะสามารถประมวลผลได้ทันที ตัวละครก็จะเคลื่อนที่ไปตามมือผู้เล่น ซึ่งจากการทดลองพบว่าในการเล่นเกมในสถานที่ที่เหมาะสม ทำให้การควบคุมตัวละครสามารถเป็นไปอย่างราบรื่น ไม่มีการรบกวนใดๆเกิดขึ้น

2. ทดสอบเงื่อนไขของเกม ในการออกแบบเกมมีเงื่อนไขว่าถ้าผู้เล่นยิงยานอวกาศฝ่ายตรงข้ามได้จะได้คะแนนเพิ่ม โดยที่กระสุนของผู้เล่นจะยิงโดยอัตโนมัติ ถ้าหากผู้เล่นถูกระเบิดของฝ่าย

ตรงข้ามหรือชนกับฝ่ายตรงข้าม Lives จะลดลงทีละ 1 ถ้า Lives เท่ากับ 0 คือจบเกม ผู้เล่นสามารถเล่นอีกครั้งได้โดยที่ คะแนนสูงสุดจะเท่ากับคะแนนที่ทำได้สูงสุดในขณะเล่น และ Lives จะเท่ากับ 5, คะแนนจะเท่ากับ 0 อีกครั้ง ถ้าหากผู้เล่นเลือกเมนู Exit โปรแกรมจะปิดทันที

จากการทดลองที่กล่าวมาข้างต้นแสดงผลดังต่อไปนี้

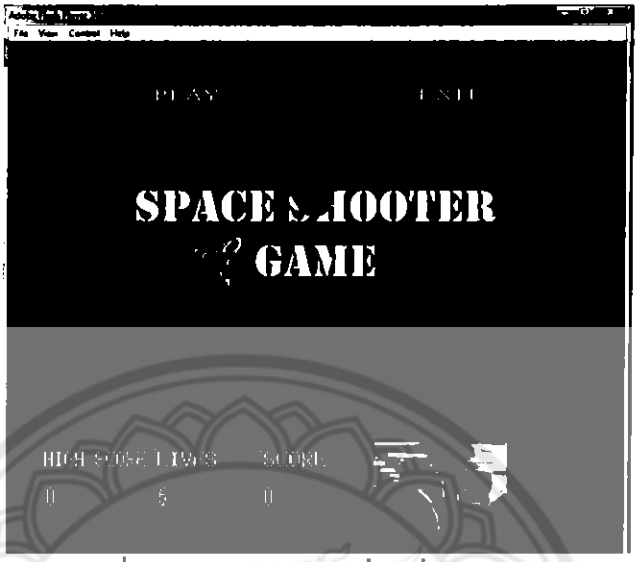
- เมื่อเปิดโปรแกรมขึ้นมา โปรแกรมจะแสดง Privacy Dialog Box เพื่อให้ผู้ใช้ทราบว่ามีการใช้งานกล้องเว็บแคม และขออนุญาตจากผู้ใช้ว่าจะยอมให้มีการเชื่อมต่อกับกล้องเว็บแคมหรือไม่ ดังรูปที่ 4.5

รูปที่ 4.5 ผลลัพธ์เมื่อเปิดโปรแกรม

- เมื่อคลิก Allow อนุญาตให้ติดต่อกล้องเว็บแคม กล้องเว็บแคมจะเริ่มทำงานและจับภาพ ดังรูปที่ 4.6

รูปที่ 4.6 ผลลัพธ์เมื่อเปิดกล้องเว็บแคม

- เมื่อผู้เล่นเกมนำมือเข้ามาในตำแหน่งที่ตั้งกล้องไว้ Symbol hand จะเข้ามาอยู่ในตำแหน่งที่มือเคลื่อนที่ทันทีดังรูปที่ 4.7



รูปที่ 4.7 Symbol hand เคลื่อนที่ไปตามมือ

- เมื่อเคลื่อนมือไปที่ปุ่ม Play จะเข้าสู่การเริ่มต้นเกม Symbol hand จะถูกซ่อนไว้แล้ว Symbol OurShip จะถูกแสดงออกมา ผู้เล่นเกมสามารถใช้มือบังคับการเคลื่อนที่ได้ทั้งแนวแกน X และแกน Y รอบพื้นที่เกมดังรูปที่ 4.8

รูปที่ 4.8 การใช้มือบังคับตัวละครในเกม

- เมื่อจบเกม OurShip จะถูกซ่อน แล้ว hand จะเข้ามาแทนที่ ผู้เล่นสามารถใช้มือในการเลือกเมนูได้อีกครั้ง

รูปที่ 4.9 Game Over

- จากการทดลองที่ผ่านมาพบว่าโปรแกรมเป็นไปตามที่ออกแบบไว้และเงื่อนไขต่างๆของเกมถูกต้อง ไม่พบข้อผิดพลาดใดๆ

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงงานนี้พัฒนาขึ้นเพื่อศึกษาและทดลองเกี่ยวกับอัลกอริทึมในการตรวจจับและติดตามการเคลื่อนที่ของมือ เพื่อนำมาใช้ในการอินเตอร์เฟสกับเกมให้สามารถควบคุมการเคลื่อนไหวของตัวละครในเกมได้ โดยมีการรับภาพจากกล้องเว็บแคมเพื่อนำมาประมวลผลจากนั้น ปรับปรุงภาพวีดีโอให้มีสีที่คมชัดขึ้น จากนั้นหาค่าสีที่ใกล้เคียงกับมือและกำหนดให้ติดตามสีนั้น โดยโครงงานนี้เป็นการประยุกต์ใช้อัลกอริทึมจากโปรแกรม ColorTracking และออกแบบเกมให้เหมาะสมกับอัลกอริทึมที่นำมาใช้

จากการทดสอบในบทที่ผ่านมาซึ่งได้ทำการทดสอบ 2 ส่วนคือ การทดสอบความแม่นยำในการประมวลผล และการทดสอบเกม

- การทดสอบความแม่นยำในการประมวลผล

ในการทดสอบความแม่นยำในการประมวลผลได้ใช้กล้องเว็บแคมความละเอียด 5M ภาพที่ได้เพียงพอต่อการรับข้อมูลภาพ แต่ถ้าใช้กล้องที่มีความละเอียดสูงกว่านี้จะทำให้ได้ภาพที่คมชัดมากขึ้นซึ่งมีผลดีต่อการประมวลผลภาพ ในการประมวลผลภาพวีดีโอนั้น มีปัจจัยหลายๆอย่างที่ทำให้การบังคับด้วยมือไม่แม่นยำ เช่นแสงสว่าง และสิ่งที่มีสีเดียวกับมือเช่นใบหน้า จะมีผลในการควบคุมตัวละครในเกมทำให้เกิดความผิดพลาดได้

- ผลการทดสอบเกม

ในการทดสอบเกมถ้าหากเล่นเกมในสภาพแวดล้อมที่เหมาะสม การเคลื่อนไหวของตัวละครจะเป็นไปอย่างราบรื่น และจากการทดสอบเล่นเกมกรากระทำต่างๆถูกต้องตามเงื่อนไขที่ได้ออกแบบไว้

5.2 ปัญหาและแนวทางแก้ไข

1. ปัญหาสภาพแวดล้อม เกมนี้ใช้กล้องเว็บแคมเป็นอุปกรณ์รับภาพ ดังนั้นผู้เล่นต้องอยู่ในสภาพแสงที่เหมาะสม คือ แสงต้องไม่สว่างหรือมืดจนเกินไป ฉากหลังควรมีสีขาวและไม่ควรมีสิ่งของหรือสิ่งอื่นๆ ปะปนอยู่ในภาพที่กล้องเว็บแคมกำลังจับภาพในขณะเล่น

2. ปัญหาการประมวลผลที่ไม่แม่นยำ ในการตรวจจับสีของมือ เนื่องจากการประมวลผลโดยการตรวจจับสีซึ่งสีของผิวมนุษย์จะได้สีเดียวกัน จึงเกิดความผิดพลาดขึ้น ถ้าหากกล้องเว็บแคมอยู่ในตำแหน่งที่มองเห็นใบหน้า จะทำให้ไม่สามารถตรวจจับมือได้แม่นยำ ตัวละครจะส่ายไปมา

ดังนั้นควรตั้งกล้องในระดับที่มองไม่เห็นหน้า แต่หากเป็นกล้องที่ติดกับคอมพิวเตอร์ จะไม่สามารถปรับระดับได้แต่สามารถเล่นโดยใช้ศีรษะบังคับได้ หรืออาจจะเยื้องตัวออกมาจากคอมพิวเตอร์ได้

5.3 ข้อเสนอแนะ/แนวทางการพัฒนาต่อ

1. ในการค้นหามือสามารถศึกษา อัลกอริธึมที่ใช้ในการค้นหาได้แม่นยำมากขึ้น เพื่อลดข้อผิดพลาดในการตรวจจับและเพิ่มความราบรื่นในการเล่นเกม

2. พัฒนาให้สามารถตรวจจับภาพจากกล้องวีดีโอเฉพาะจุดที่ต้องการได้

3. พัฒนาเกมให้มีรูปแบบที่มีการเคลื่อนไหวร่างกายมากขึ้น เพื่อเพิ่มการปฏิสัมพันธ์กับเกมและใช้ในการส่งเสริมการออกกำลังกายได้ด้วย

มหาวิทยาลัยนเรศวร

เอกสารอ้างอิง

[1] บุญเลิศ อรุณพิบูลย์. **"เริ่มต้นกับ Flash."**[Online].Available : http://www.stks.or.th/web/index.php?option=com_docman&task=doc_download &gid =98&Itemid=31 2009.

[2] **"คอมพิวเตอร์วิทัศน์ (Computer Vision)."**[Online].Available : http://www.narisa.com/forums/index.php?app=blog& module=display&section=blog &blogid=17&showentry=1946 2009.

[3] วิกิพีเดีย สารานุกรมเสรี. **"Video processing."**[Online].Available : http://th.asiaonline.com/en/article?article=Multiplexed#Video_processing 2009.

[4] ดร.พนมขวัญ ริยะมงคล. **"Image Processing."**[Online].Available : http://www.ecpe.nu.ac.th/panomkhawn/imagepro/pdf/ch01.pdf 2010.

[5] Yilmaz, A., Javed, O., and Shah, M. **"Object tracking."**[Online].Available : http://portal.acm.org/citation.cfm?id=1177355 2011.

[6] รศ. ดร. มนตรี กาญจนะเดชะ. **"ระบบสี RGB."**[Online].Available : http://fivedots.coe.psu.ac.th/~montri/Teaching/image/chap1.htm2011.

[7] Jonathan, C. **"Color Tracking."**[Online].Available : http://redblind.com/blog/2009/09/09/color-tracking/2011.

[8] Jason, T.**"AS3 Flash Games for Beginners."**[Online].Available : http://asgamer.com/2009/flash-game-design-basics-adding-library-objects-to-stage-with-as3 2011.

[9] ชัชวาลย์ ฐิติพจน์. "3 เทคนิคการจัดองค์ประกอบ."[Online].Available : http://202.28.25.135/~chatchawan/website_design_page2.html 2011.

ประวัติผู้เขียนโครงงาน

ชื่อ นายภากร น้อยสอน
ภูมิลำเนา 107/1 ม.6 ต.บางไผ่ อ.บางมูลนาก จ.พิจิตร 66120
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนตะพานหิน
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : tarkunno191@hotmail.com

ชื่อ นางสาวสุภาวดี พาพิมพ์
ภูมิลำเนา 286/6 ม.4 ต.แสนตอ อ.น้ำปาด จ.อุตรดิตถ์ 53110
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนน้ำปาดชนูปถัมภ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : soft_works@hotmail.com