การศึกษาการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้กระบวนการตัดสินใจมาร์คอฟ

A STUDY OF ROBOT PATH FINDING USING MARKOV DECISION PROCESS

นางสาวชญานิษฐ์ หวังวิริยะพันธ์
นางสาวภัทราภรณ์ นาควรสุข
MS.CHAYANIT WANGVIRIYAPHAN
MS.PATTARAPORN NAKWORASUK

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A STUDY OF ROBOT PATH FINDING USING MARKOV DECISION PROCESS

MS.CHAYANIT WANGVIRIYAPHAN
MS.PATTARAPORN NAKWORASUK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ การศึกษาการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้กระบวนการตัดสินใจมาร์คอฟ

A STUDY OF ROBOT PATH FINDING USING MARKOV DECISION PROCESS

นักศึกษา

นางสาวชญานิษฐ์ หวังวิริยะพันธ์	รหัสประจำตัว	52010200
นางสาวภัทราภรณ์ นาควรสุข	รหัสประจำตัว	52010898

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(ดร.อุดม จันทร์จรัสสุข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การศึกษาการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้กระบวนการตัดสินใจมาร์คอฟ
นักศึกษา	นางสาวชญานิษฐ์ หวังวิริยะพันธ์ นางสาวภัทราภรณ์ นาควรสุข
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2555
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ดร.อุดม จันทร์จรัสสุข

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ มีวัตถุประสงค์เพื่อศึกษา และประยุกต์ใช้กระบวนการตัดสินใจมาร์คอฟ ในการค้นหาเส้นทางเดินของหุ่นยนต์ จากตำแหน่งเริ่มต้นไปยังตำแหน่งเป้าหมายที่ไม่ทราบล่วงหน้า โดยหลบหลีกสิ่งกีดขวางที่ถูกวางไว้ การศึกษานี้มีขั้นตอนการดำเนินงานเริ่มจาก การศึกษาความเป็นไปได้ของการนำทฤษฎีของมาร์คอฟมาประยุกต์ใช้ โดยการพิจารณาค่าความน่าจะเป็นของการเลือกทิศทางการเคลื่อนที่ จากตำแหน่งปัจจุบันไปยังตำแหน่งถัดไป หรือการปรับเปลี่ยนสถานการณ์เคลื่อนที่ของหุ่นยนต์ รวมทั้งได้มีการศึกษาถึงปัจจัยที่มีผลต่อการปรับเปลี่ยนสถานะ และการจัดทำโปรแกรมคอมพิวเตอร์สำหรับจำลองค้นหาเส้นทางเดินของหุ่นยนต์ จากการทดสอบโปรแกรม โดยการจำลองการเคลื่อนที่ของหุ่นยนต์บนคอมพิวเตอร์ในสนามจำลองขนาด 8x8 พบว่าตัวแปรที่มีผลต่อจำนวนการเคลื่อนที่ของหุ่นยนต์ คือ จำนวนสิ่งกีดขวาง และการเปลี่ยนน้ำหนักที่ให้กับค่าความน่าจะเป็นของแต่ละทิศทางที่จะเคลื่อนที่ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	A STUDY OF ROBOT PATH FINDING USING MARKOV DECISION PROCESS
Student	Ms. Chayanit Wangviriyaphan Ms. Pattaraporn Nakworasuk
Degree	Bachelor of Engineering in Industrial Engineering King Mongkut's Institute of Technology Ladkrabang
Academic Year	2012
Thesis Advisor	Dr. Udom Janjarassuk

ABSTRACT

The objective of this thesis is to study and apply the Markov decision process in order to find the moving path of a robot from a starting position to an unknown goal position by avoiding the obstructions. The study starts from studying the possibility of using the Markov Theory for finding path of the robot by considering the probability of changing moving direction from the current position to the next position, or the state transition of the moving robot. Furthermore, the factors which affect the state transition are also studied. In addition, we developed a computer program to simulate the movement of robot on a 8x8 rectangular grid. From the experiment, the factors that affect the number of movement of the robot are the number of obstructions and the weight of probability of changing direction.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่องศึกษาการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้กระบวนการตัดสินใจมาร์คอฟ สามารถสำเร็จลุล่วงไปได้ด้วยดีด้วยความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ซึ่งผู้มีพระคุณท่านแรกที่คณะผู้จัดทำใคร่ขอกราบขอบพระคุณคือ ดร.อุดม จันทร์จรัสสุข อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ได้ให้ทั้งความรู้ คำแนะนำ ความช่วยเหลือ และความเอาใจใส่ในทุกๆด้านตลอดเวลาที่ผ่านมา

กราบขอบพระคุณ รศ.ดร.ฤดี มาสุจันท์ และ ผศ.ดร. สกนธ์ คล่องบุญจิต คณะกรรมการสอบความคืบหน้าของปริญญานิพนธ์ ที่ได้ให้ข้อคิดเห็นและข้อเสนอแนะตั้งแต่เริ่มต้น ทำให้ปริญญานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณภาควิชาอุตสาหการ และเพื่อนๆนักศึกษาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง รุ่น 13 สำหรับกำลังใจดีๆ ในการทำปริญญานิพนธ์

สุดท้ายนี้ขอขอบพระคุณพ่อและแม่ พร้อมทั้งเหล่าคณาจารย์ทุกท่าน ที่ได้ให้กำลังใจ วิชาความรู้ และความช่วยเหลือในทุกๆด้าน จนกระทั่งปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงลงด้วยดี

นางสาวชญานิษฐ์ หวังวิริยะพันธ์
นางสาวภัทราภรณ์ นาควรสุข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของปริญญานิพนธ์ 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวกับลูกโซ่มาร์คอฟ 3

2.2 วิธีการคำนวณค่าความน่าจะเป็นของการเคลื่อนที่ 5

2.3 การจำลองการเคลื่อนที่ 7

2.4 การคำนวณค่าเฉลี่ยตัวอย่าง 7

2.5 การคำนวณค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง 8

2.6 การประมาณค่า 8

2.6.1 ทฤษฎีที่เกี่ยวกับการประมาณค่าเฉลี่ยประชากรเดียวแบบช่วง 9

2.6.2 ทฤษฎีที่เกี่ยวกับการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง 9

บทที่ 3 วิธีการดำเนินงาน

3.1 การศึกษาความเป็นไปได้ของการนำทฤษฎีของมาร์คอฟ เพื่อใช้ค้นหาเส้นทางเดินของหุ่นยนต์ 11

3.2 การศึกษาปัจจัยที่มีผลต่อการปรับเปลี่ยนสถานะ 17

3.3 การจัดทำโปรแกรมคอมพิวเตอร์สำหรับจำลองการค้นหาเส้นทางเดินของหุ่นยนต์ 20

3.3.1 การเขียนผังงาน 21

3.3.2 การออกแบบโปรแกรม 29

3.3.3 การเขียนโปรแกรม 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3.4 การทดสอบและแก้ไขโปรแกรม 32

3.3.5 การนำโปรแกรมไปใช้งาน 32

บทที่ 4 ผลการดำเนินงาน

4.1 การทดลองเพื่อหาจำนวนรอบในการเดิน 33

4.2 การทดลองเปลี่ยนจำนวนสิ่งกีดขวาง 38

4.3 การทดลองเปลี่ยนน้ำหนักที่ให้กับค่าความน่าจะเป็น 40

บทที่ 5 สรุปผลการวิจัย

5.1 สรุปผลการดำเนินงาน 44

5.2 สรุปผลการนำโปรแกรมที่พัฒนาขึ้นมาใช้ทดลองเพื่อศึกษาตัวแปร ที่มีผลต่อการเคลื่อนที่ 45

5.2.1 การทดลองเปลี่ยนจำนวนสิ่งกีดขวาง 45

5.2.2 การทดลองเปลี่ยนน้ำหนักที่ให้กับความน่าจะเป็น 46

5.3 แนวทางในการนำไปพัฒนา 46

หนังสืออ้างอิง 47

ภาคผนวก ก ..... ผก 1

ภาคผนวก ข ..... ผข 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ให้มีช่องทางที่สามารถไปยังเป้าหมายได้ 12
ตารางที่ 3.2 แสดงการคำนวณในโปรแกรม Matlab เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ให้มีช่องทาง ที่สามารถไปยังเป้าหมายได้ 13
ตารางที่ 3.3 แสดงเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้ 15
ตารางที่ 3.4 แสดงการคำนวณในโปรแกรม Matlab เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทาง ที่สามารถไปยังเป้าหมายได้ 15
ตารางที่ 3.5 แสดงเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ เมื่อให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า 17
ตารางที่ 3.6 แสดงการคำนวณในโปรแกรม Matlab เมื่อให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า 18
ตารางที่ 4.1 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยใช้ 125 รอบ 34
ตารางที่ 4.2 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยใช้สิ่งกีดขวาง 10 ชิ้น 38
ตารางที่ 4.3 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยใช้สิ่งกีดขวาง 15 ชิ้น 39
ตารางที่ 4.4 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด 41
ตารางที่ 4.5 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยให้น้ำหนักใน 2 ทิศทาง ที่จะไปยังเป้าหมายมีค่ามากกว่า 42
ตารางที่ 5.1 แสดงการสรุปค่าของเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ 45
ตารางที่ 5.2 แสดงค่าเฉลี่ยตัวอย่างและส่วนเบี่ยงเบนมาตรฐานตัวอย่าง ในการทดลองเปลี่ยนจำนวนสิ่งกีดขวาง 45
ตารางที่ 5.3 แสดงค่าเฉลี่ยตัวอย่างและส่วนเบี่ยงเบนมาตรฐานตัวอย่าง ในการทดลองเปลี่ยนน้ำหนัก ที่ให้กับความน่าจะเป็น 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 แสดงตำแหน่งจุดเริ่มต้น เป้าหมาย และสิ่งกีดขวาง สำหรับตัวอย่างการหาค่าความน่าจะเป็นของการเคลื่อนที่ 6

รูปที่ 2.2 แสดงตำแหน่งจุดเริ่มต้น สำหรับตัวอย่างการหาค่าความน่าจะเป็นของการเคลื่อนที่ 6

รูปที่ 3.1 แสดงตำแหน่งจุดเริ่มต้น เป้าหมาย และสิ่งกีดขวางมี โดยมีช่องทางที่สามารถไปยังเป้าหมายได้ 11

รูปที่ 3.2 แสดงตำแหน่งจุดเริ่มต้น เป้าหมาย และสิ่งกีดขวาง โดยไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้ 14

รูปที่ 3.3 แสดงขั้นตอนการพัฒนาโปรแกรม 20

รูปที่ 3.4 แสดงผังงานการทำงานของโปรแกรมรวม 21

รูปที่ 3.5 แสดงผังงานการคำนวณค่าความน่าจะเป็นของการเคลื่อนที่ 24

รูปที่ 3.6 แสดงผังงานการเลือกทิศทางในการเคลื่อนที่ 25

รูปที่ 3.7 แสดงการจำลองการเคลื่อนที่ 26

รูปที่ 3.8 แสดงการคำนวณค่าเฉลี่ยตัวอย่าง 27

รูปที่ 3.9 แสดงการคำนวณค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง 28

รูปที่ 3.10 แสดงส่วนรับข้อมูลในฟอร์มที่ 1 29

รูปที่ 3.11 แสดงส่วนรับข้อมูลในฟอร์มที่ 2 30

รูปที่ 3.12 แสดงส่วนแสดงผล 31

รูปที่ 4.1 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยใช้ 125 รอบ 34

รูปที่ 4.2 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 125 รอบ 34

รูปที่ 4.3 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 250 รอบ 35

รูปที่ 4.4 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 500 รอบ 36

รูปที่ 4.5 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 1000 รอบ 36

รูปที่ 4.6 แสดงช่วงความเชื่อมั่น 95% ของ μ 37

รูปที่ 4.7 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยใช้สิ่งกีดขวาง 10 ชิ้น 38

รูปที่ 4.8 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง โดยใช้สิ่งกีดขวาง 10 ชิ้น 39

รูปที่ 4.9 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยใช้สิ่งกีดขวาง 15 ชิ้น 39

รูปที่ 4.10 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง โดยใช้สิ่งกีดขวาง 15 ชิ้น 40

รูปที่ 4.11 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด 41

รูปที่ 4.12 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง โดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 4.13 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย
โดยให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า..... 42
รูปที่ 4.14 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง
โดยให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า..... 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันได้เกิดเหตุการณ์และอุบัติภัยต่างๆมากมาย ที่ทำให้เกิดความเสียหายแก่ชีวิต และทรัพย์สินของมนุษย์ ไม่ว่าจะเป็น เหตุการณ์ตึกถล่ม แผ่นดินไหว เหตุเพลิงไหม้ เป็นต้น ล้วนแล้วแต่ทำให้เกิดพื้นที่ หรือสภาพแวดล้อมที่เข้าถึงยากลำบาก ซึ่งสภาพแวดล้อมดังกล่าวอาจทำให้เกิดอันตรายแก่ผู้ที่เข้าไปทำการค้นหาผู้ประสบภัย หรือต้องเสียเวลานานในการค้นหา ทำให้การเข้าไปช่วยเหลือผู้ประสบภัยยังต้องมีการพึ่งพาเทคโนโลยี เช่น การสร้างหุ่นยนต์ที่มีขนาดเล็กสามารถเข้าถึงผู้ประสบภัย และผ่านสิ่งกีดขวางได้ง่าย ทำให้ลดความเสี่ยงของผู้ที่เข้าไปทำการค้นหา โดยจะมีการส่งหุ่นยนต์เข้าไปค้นหาผู้ประสบภัยก่อน เพื่อหาตำแหน่งที่แน่นอน ทำให้สามารถเข้าไปช่วยเหลือได้ทันท่วงที จึงเห็นได้ว่าการใช้หุ่นยนต์ เป็นเรื่องที่สำคัญในการเข้าช่วยเหลือ และในการเข้าช่วยเหลือ ในพื้นที่อาจจะมีสิ่งกีดขวาง หรือซากปรักหักพังต่างๆที่ไม่ทราบล่วงหน้าระหว่างเส้นทางของการเข้าไปช่วยเหลือ ทำให้เป็นอุปสรรคต่อการค้นหาเส้นทาง และเป็นการยากต่อการตัดสินใจในการเลือกเส้นทาง ทำให้ต้องมีกระบวนการตัดสินใจเข้ามาเกี่ยวข้อง

มาร์คอฟ เป็นกระบวนการตัดสินใจหนึ่ง โดยอาศัยหลักการที่สถานะในอนาคตของกระบวนการจะขึ้นอยู่กับสถานะปัจจุบันเพียงอย่างเดียว จึงสามารถนำมาร์คอฟมาประยุกต์ใช้งาน เช่น ปัญหาด้านการผลิต ที่นำมาประมาณว่าโอกาสที่เครื่องจักรจะใช้งานได้ดีในสัปดาห์หน้าหรือไม่ หรือการตัดสินใจในเรื่องของการตลาด ที่นำมาประมาณโอกาสที่ลูกค้าที่เคยซื้อยี่ห้อ A จะเปลี่ยนไปซื้อยี่ห้อ B แทน เป็นต้น

นอกจากนั้น ในการเลือกทิศทางของหุ่นยนต์ ถือได้ว่าเป็นการตัดสินใจอย่างหนึ่ง ดังนั้นจึงเห็นว่าการนำกระบวนการตัดสินใจมาร์คอฟ มาประยุกต์ใช้กับการค้นหาเส้นทางเดินของหุ่นยนต์เป็นเรื่องที่น่าสนใจ เพราะค่อนข้างใหม่หรือมีงานวิจัยที่เกี่ยวข้องน้อย เนื่องจากผู้ที่สนใจทางด้านการทำหุ่นยนต์ ส่วนใหญ่จะทำการศึกษาไปในทางของการควบคุมมากกว่า เช่น การควบคุมระบบไฟฟ้า ระบบแมคคาทรอนิกส์ เป็นต้น

1.2 วัตถุประสงค์

1. เพื่อประยุกต์ใช้กระบวนการตัดสินใจมาร์คอฟ ในการค้นหาเส้นทางเดินของหุ่นยนต์
2. เพื่อพัฒนาโปรแกรมคอมพิวเตอร์ ในการจำลองการค้นหาเส้นทางเดินของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

1. พัฒนาโปรแกรมในการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้ภาษา Visual C #
2. ทดสอบโปรแกรม โดยการจำลองการเคลื่อนที่ของหุ่นยนต์บนคอมพิวเตอร์ ในสนามจำลองขนาด 8x8
3. ทำการทดลองโปรแกรม เพื่อศึกษาตัวแปรที่มีผลต่อจำนวนการเคลื่อนที่ของหุ่นยนต์
4. ผู้ใช้สามารถกำหนดตำแหน่งเริ่มต้น และตำแหน่งเป้าหมายล่วงหน้าไว้แน่นอนเพื่อใช้ในการทดลอง โดยที่ผู้ใช้สามารถเปลี่ยนตำแหน่งเองได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเป็นสื่อการเรียนรู้ เพื่อให้เข้าใจเกี่ยวกับกระบวนการตัดสินใจมาร์คอฟ
2. เพื่อเป็นโปรแกรมต้นแบบ สำหรับนำไปพัฒนา เพื่อประยุกต์ใช้กับหุ่นยนต์จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

การพัฒนาโปรแกรมคอมพิวเตอร์ โดยการประยุกต์ใช้กระบวนการตัดสินใจมาร์คอฟ มาใช้ในการค้นหาเส้นทางเดินของหุ่นยนต์ จึงได้มีการอธิบายทฤษฎีที่เกี่ยวข้องกับลูกโซ่มาร์คอฟ นอกจากนั้นได้มีการอธิบายทฤษฎีที่เกี่ยวข้องกับการเคลื่อนที่ของหุ่นยนต์บนคอมพิวเตอร์ โดยอาศัยทฤษฎีการคำนวณค่าความน่าจะเป็นของการเคลื่อนที่ และทฤษฎีการจำลองการเคลื่อนที่ ส่วนในการทดลองเพื่อศึกษาตัวแปรที่มีผลต่อจำนวนการเคลื่อนที่ อธิบายโดยอาศัยทฤษฎีการคำนวณค่าเฉลี่ยตัวอย่าง และค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง เพื่อนำมาใช้ในทฤษฎีที่เกี่ยวกับการประมาณค่า

2.1 ทฤษฎีที่เกี่ยวกับลูกโซ่มาร์คอฟ

ลูกโซ่มาร์คอฟ (Markov Chain) คือ กระบวนการสุ่มที่มีคุณสมบัติมาร์คอฟ (Markov Property) ซึ่งก็คือ สถานะในอนาคตของกระบวนการจะไม่ขึ้นอยู่กับสถานะในอดีต แต่จะขึ้นอยู่กับสถานะปัจจุบันเพียงอย่างเดียว คุณสมบัติดังกล่าวทำให้การวิเคราะห์ทำได้ง่ายขึ้น เนื่องจากไม่จำเป็นต้องทราบการเปลี่ยนแปลงทั้งวัฏจักร เพียงแค่รู้ถึงสถานะปัจจุบันก็เพียงพอ

บ่อยครั้งที่รูปแบบของลูกโซ่มาร์คอฟ ถูกใช้อธิบายความหมายของกระบวนการมาร์คอฟ (Markov Process) ที่มีปริภูมิสถานะ (State Space) หรือค่าที่เป็นไปได้ทั้งหมดของ X_n เป็นแบบจำนวนจำกัด หรือนับได้ (Discrete) และโดยปกติแล้วลูกโซ่มาร์คอฟ ถูกกำหนดให้เวลาเป็นแบบไม่ต่อเนื่อง ซึ่งในปริญญานิพนธ์ฉบับนี้ จะเกี่ยวข้องกับเนื้อหาในส่วนของเรื่องลูกโซ่มาร์คอฟแบบเวลาไม่ต่อเนื่อง (Discrete Time Markov Chain)

จากการพิจารณาถึงกระบวนการสโตคาสติก (Stochastic Process) $\{X_n, \mathrm{n} = 0,1,2,\dots\}$ ซึ่ง กระบวนการสโตคาสติก เป็นที่รู้จักในชื่อของลูกโซ่มาร์คอฟ โดยค่าที่เป็นไปได้ของกระบวนการสโตคาสติก จะเขียนแทนด้วยเซ็ทของจำนวนเต็มบวก $\{0,1,2,\dots\}$ ถ้า $X_n = i$ กระบวนการที่กล่าวถึงจะอยู่ที่สถานะ i ณ เวลา n ซึ่งกระบวนนี้จะเรียกว่าเป็นลูกโซ่มาร์คอฟได้ ก็ต่อเมื่อ ถ้าที่สถานะ i ใดๆมีค่าความน่าจะเป็นที่แน่นอน โดย P_{ij} เป็นค่าความน่าจะเป็นที่จะเปลี่ยนจากสถานะ i ไปยังสถานะ j ดังนั้นทางคณิตศาสตร์กล่าวได้ว่า สำหรับทุกๆสถานะ $i_0, i_1, \dots, i_{n-1}, i, j$ และทุกๆค่าของ $n \geq 0$ ดังแสดงในสมการที่ 2.1 (Freedman, 1970)

$$P\{X_{n+1} = j | X_n = i, X_{n-1} = i-1, \dots, X_1 = i_1, X_0 = i_0\} = P_{ij} \qquad (2.1)$$

จากสมการที่ (2.1) ให้สถานะในอนาคต คือ X_{n+1} ให้สถานะในอดีต คือ $X_0, X_1, \dots X_{n-1}$ และสถานะปัจจุบัน คือ X_n

โดยที่ P_{ij} คือ ความน่าจะเป็นของการเปลี่ยนสถานะ (Transition Probability) ซึ่งจะต้องมีคุณสมบัติดังแสดงในสมการที่ 2.2 (Freedman, 1970)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_{ij} \geq 0 \quad i,j \geq 0 \qquad \sum_{j=0}^{\infty} P_{ij} = 1 \qquad i = 0,1, \ldots. \tag{2.2}$$

ให้ P เขียนแทนด้วยเมตริกซ์ ของความน่าจะเป็นของการเปลี่ยนสถานะเพียงครั้งเดียว จากสถานะ i ไปยังสถานะ j (one - Step Transition of Probabilities, P_{ij}) ดังแสดงในสมการที่ 2.3 (Freedman, 1970)

$$P = \begin{Vmatrix} P_{00} & P_{01} & P_{02} & \ldots \\ P_{10} & P_{11} & P_{12} & \ldots \\ P_{20} & P_{21} & P_{22} & \ldots \\ P_{30} & P_{31} & P_{32} & \ldots \end{Vmatrix} \tag{2.3}$$

ตัวอย่างที่ 2.1

สมมุติให้โอกาสที่ฝนจะตกในวันพรุ่งนี้ ขึ้นอยู่กับสภาพอากาศก่อนหน้าที่ผ่านมาเท่านั้น และไม่ขึ้นอยู่กับสภาพอากาศในอดีต สมมติถ้าวันนี้ฝนตก จะทำให้ฝนตกในวันพรุ่งนี้ด้วยความน่าจะเป็น α และถ้าวันนี้ฝนไม่ตก จะทำให้ฝนตกในวันพรุ่งนี้ด้วยความน่าจะเป็น β

ถ้าเราให้กระบวนการอยู่ในสถานะที่ 0 เมื่อฝนตก และอยู่ในสถานะที่ 1 เมื่อฝนไม่ตก ดังนั้น 2 สถานะของลูกโซ่มาร์คอฟ (Markov Chain) จะมีความน่าจะเป็นของการเปลี่ยนสถานะ ดังนี้

$$P = \begin{Vmatrix} \alpha & 1-\alpha \\ \beta & 1-\beta \end{Vmatrix}$$

หลังจากที่ได้กล่าวถึง ความน่าจะเป็นของการเปลี่ยนสถานะเพียงครั้งเดียว จากสถานะ i ไปยังสถานะ j (One - Step Transition of Probabilities, P_{ij}) ไปแล้ว ต่อไปจะเป็นการกล่าวถึง ความน่าจะเป็นของการเปลี่ยนสถานะเป็นจำนวน n ครั้ง จากสถานะ i ไปยังสถานะ j (n – Step Transition of Probabilities , P_{ij}^{n}) ดังแสดงในสมการที่ 2.4

$$P_{ij}^{n} = P\{X_{n+k} = j | X_k = i\} \qquad n \geq 0 \quad i,j \geq 0 \tag{2.4}$$

แน่นอนที่ $P_{ij}^{1} = P_{ij}$ โดยที่ค่า P_{ij}^{n} สามารถคำนวณได้จากสมการของ Chapman-Kolmogorov ดังแสดงในสมการที่ 2.5

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P_{ik}^{n} P_{kj}^{m} \qquad n,m \geq 0 \; ; \; i,j \geq 0 \tag{2.5}$$

ถ้าเราให้ $P^{(n)}$ แสดงถึงเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะเป็นจำนวน n ครั้ง จากสถานะ i ไปยังสถานะ j ดังแสดงในสมการที่ 2.6

$$P^{(n+m)} = P^{(n)} \cdot P^{(m)} \tag{2.6}$$

ซึ่ง • แสดงถึงการคูณของเมตริกซ์ ดังแสดงในสมการที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P^{(2)} = P^{(1+1)} = P \cdot P = P^2 \tag{2.7}$$

จะเห็นได้ว่า เมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะเป็นจำนวน n ครั้ง จากสถานะ i ไปยังสถานะ j บางครั้งได้มาจากการคูณเมตริกซ์ P ด้วยตัวเองเป็นจำนวน n ครั้ง

ตัวอย่างที่ 2.2

พิจารณาตัวอย่างที่ 1 ซึ่งสภาพอากาศถูกพิจารณาเป็น 2 สถานะ ถ้า α = 0.7 และ β = 0.4 การหาความน่าจะเป็นที่ฝนจะตก 4 วันต่อจากวันนี้ โดยกำหนดให้วันนี้ฝนตก สามารถทำได้โดย

ความน่าจะเป็นของการเปลี่ยนสถานะเพียงครั้งเดียว จากสถานะ i ไปยังสถานะ j เป็นดังนี้

ดังนั้น

$$P = \begin{Vmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{Vmatrix}$$

$$P^{(2)} = P^2 = \begin{Vmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{Vmatrix} \cdot \begin{Vmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{Vmatrix}$$

$$= \begin{Vmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{Vmatrix}$$

$$P^{(4)} = (P^2)^2 = \begin{Vmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{Vmatrix} \cdot \begin{Vmatrix} 0.61 & 0.39 \\ 0.52 & 0.48 \end{Vmatrix}$$

$$= \begin{Vmatrix} 0.5749 & 0.4251 \\ 0.5668 & 0.4332 \end{Vmatrix}$$

และความน่าจะเป็นที่ต้องการ P_{00}^4 เท่ากับ 0.5749

2.2 วิธีการคำนวณค่าความน่าจะเป็นของการเคลื่อนที่

ความน่าจะเป็นของการเคลื่อนที่ หรือความน่าจะเป็นของแต่ละทิศทาง คือ ฟังก์ชันความน่าจะเป็นของตัวแปรสุ่ม X (Probability Distribution Function: PDF) โดยที่ X มีค่าเป็น 1, 2, 3 และ 4 ซึ่งจะใช้แทนการเคลื่อนที่ไปด้านบน, ล่าง, ซ้าย และขวาตามลำดับ

ในการหาค่าความน่าจะเป็นของการเคลื่อนที่ จะเกี่ยวข้องกับค่าน้ำหนักที่ผู้ใช้กำหนด และสิ่งกีดขวางที่อยู่รอบๆตำแหน่งปัจจุบัน โดยสามารถคำนวณได้ดังแสดงในสมการที่ 2.8

$$P_i = \frac{w_i q_i}{\sum_{i=1}^{n} w_i q_i} \tag{2.8}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

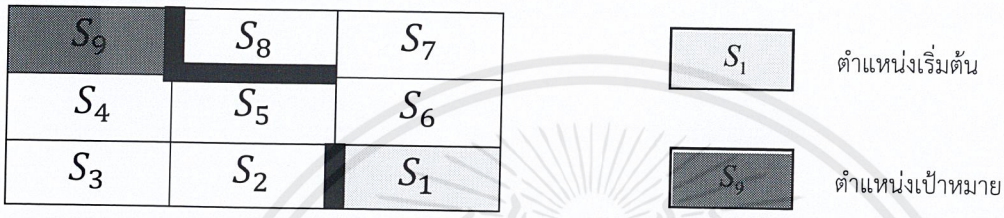
โดยที่ P_i คือ ความน่าจะเป็นของการเคลื่อนที่ โดยที่ $i = 1,2,3$ และ 4

w_i คือ ค่าน้ำหนักของความน่าจะเป็นของทิศทางการเคลื่อนที่ โดยที่ $i = 1,2,3$ และ 4

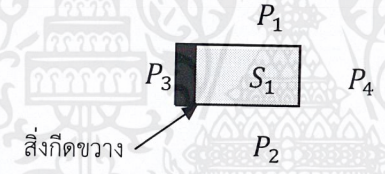
$q_i = \begin{cases} 0 \text{ ; ไม่มีสิ่งกีดขวาง} \\ 1 \text{ ; มีสิ่งกีดขวาง} \end{cases}$

n คือ จำนวนทิศทางทั้งหมด ซึ่งในที่นี้ $n = 4$ เนื่องจากมีทั้งหมด 4 ทิศทางในละตำแหน่ง คือ ด้านบน ด้านล่าง ด้านซ้าย และด้านขวา

ตัวอย่างที่ 2.3 การหาค่าความน่าจะเป็นของการเคลื่อนที่



รูปที่ 2.1 แสดงตำแหน่งจุดเริ่มต้น เป้าหมาย และสิ่งกีดขวาง สำหรับตัวอย่างการหาค่าความน่าจะเป็นของการเคลื่อนที่



รูปที่ 2.2 แสดงตำแหน่งจุดเริ่มต้น สำหรับตัวอย่างการหาค่าความน่าจะเป็นของการเคลื่อนที่

จากรูปที่ 2.1 ถ้าพิจารณาที่ตำแหน่งเริ่มต้นและตำแหน่งเป้าหมาย จะเห็นได้ว่าทิศทางที่ควรมีค่าน้ำหนักมากกว่า คือ ด้านบน และด้านซ้าย เนื่องจากทั้ง 2 ทิศทางเป็นทิศทางที่จะไปยังเป้าหมาย และจากรูปที่ 2.2 จึงสมมติให้ $w_1 = 2,\ w_2 = 1,\ w_3 = 2,\ w_4 = 1$ และ $q_1 = 1,\ q_2 = 1,\ q_3 = 0,\ q_4 = 1$ จากสมการที่ 2.8

จะได้ $\sum_{i=1}^{4} w_i q_i = (2 \times 1) + (1 \times 1) + (2 \times 0) + (1 \times 1) = 4$

และ $P_1 = \frac{2 \times 1}{4} = 0.5$

$P_2 = \frac{1 \times 1}{4} = 0.25$

$P_3 = \frac{2 \times 0}{4} = 0$

$P_4 = \frac{1 \times 1}{4} = 0.25$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การจำลองการเคลื่อนที่

สำหรับในกรณีของการจำลองการหาเส้นทางเดินของหุ่นยนต์ จะเป็นการสุ่มตัวอย่างจากการแจกแจงความน่าจะเป็น โดยตัวแปรสุ่มที่เกิดขึ้นจะเป็นตัวแปรสุ่ม X แบบไม่ต่อเนื่อง (Discrete Random Variable) เพราะมีจำนวนของตัวแปรสุ่มที่จำกัด โดยที่ X มีค่าเป็น 1, 2, 3 และ 4 ซึ่งจะใช้แทนการเคลื่อนที่ไปด้านบน, ล่าง, ซ้าย และขวา ตามลำดับ

วิธีที่จะใช้ในการสุ่มตัวอย่างจากการแจกแจงความน่าจะเป็น คือ วิธีการแปลงผกผัน (Inverse Transform Method) ซึ่งเป็นวิธีในการสร้างตัวแปรสุ่มจากฟังก์ชันการแจกแจงความน่าจะเป็นสะสมของตัวแปรสุ่ม X (Cumulative Distribution Function: CDF)

การทำวิธีการแปลงผกผัน จะต้องมีฟังก์ชันความน่าจะเป็นของตัวแปรสุ่ม X ก่อน สำหรับตัวแปรสุ่ม X ที่ค่าของตัวแปรสุ่มมีค่าเป็น 1, 2, 3 และ 4 จะได้ว่าฟังก์ชันความน่าจะเป็นของตัวแปรสุ่ม X แทนด้วยสัญลักษณ์ $f(x) = \mathrm{P(X = x)}$

การหาเส้นทางเดินของหุ่นยนต์ ค่าความน่าจะเป็นของตัวแปรสุ่ม X มีทั้งหมด 4 ค่า คือ P_1, P_2, P_3, P_4 โดยได้มาจากการคำนวณตามสมการที่ 1 จากนั้นนำค่าความน่าจะเป็นที่ได้มาคำนวณหาความน่าจะเป็นสะสม ซึ่งก็คือฟังก์ชันที่แสดงถึงค่าความน่าจะเป็นสะสมเมื่อ X มีค่าน้อยกว่าค่าใดค่าหนึ่ง ดังแสดงในสมการที่ 2.9

$$CDF = F(X) = P(X \leq x) = \sum_{i=1}^{n} P(X_i) \quad ; i = 1,2,3,4 \quad ; n = 1,2,3,4 \tag{2.9}$$

ดังนั้น CDF (1) = P_1

CDF (2) = $P_1 + P_2$

CDF (3) = $P_1 + P_2 + P_3$

CDF (4) = $P_1 + P_2 + P_3 + P_4$

ซึ่งฟังก์ชันการแจกแจงความน่าจะเป็นสะสมของตัวแปรสุ่ม X มีคุณสมบัติคือ $0 \leq F(x) \leq 1$ ดังนั้น หลังจากที่ได้ฟังก์ชันการแจกแจงความน่าจะเป็นสะสมของตัวแปรสุ่ม X มาแล้ว จะมีการสุ่มตัวเลขที่มีค่าระหว่าง 0-1 โดยกำหนดให้ตัวเลขที่สุ่ม = U เพื่อนำมาเปรียบเทียบกับค่า CDF ซึ่งค่า CDF ทั้ง 4 ค่าจะถูกแบ่งออกเป็นช่วงๆ ดังนี้

ถ้า 0 $\leq$ U $<$ CDF (1) แสดงว่าตัวแปรสุ่มที่ได้ คือ 1 ซึ่งหมายถึงการเคลื่อนที่ไปด้านบน

ถ้า CDF (1) $\leq$ U $<$ CDF (2) แสดงว่าตัวแปรสุ่มที่ได้ คือ 2 ซึ่งหมายถึงการเคลื่อนที่ไปด้านล่าง

ถ้า CDF (2) $\leq$ U $<$ CDF (3) แสดงว่าตัวแปรสุ่มที่ได้ คือ 3 ซึ่งหมายถึงการเคลื่อนที่ไปด้านซ้าย

ถ้า CDF (3) $\leq$ U $<$ CDF (4) แสดงว่าตัวแปรสุ่มที่ได้ คือ 4 ซึ่งหมายถึงการเคลื่อนที่ไปด้านขวา

2.4 การคำนวณค่าเฉลี่ยตัวอย่าง (Sample Average, $\bar{x}$)

ในที่นี้จะเป็นการเก็บข้อมูลจากตัวอย่าง ที่ใช้เสมือนเป็นตัวแทนของประชากร เนื่องจากการเก็บข้อมูลจากสมาชิกทั้งหมดของประชากรในบางครั้ง ไม่สามารถทำได้ด้วยสาเหตุต่างๆ เช่น ประชากรมีขนาดใหญ่ ทำให้การเก็บข้อมูลต้องใช้เวลานานมาก จึงไม่สามารถเก็บข้อมูลจากสมาชิกทั้งหมดได้

ดังนั้น ค่าเฉลี่ยของจำนวนก้าว คือ ค่าเฉลี่ยตัวอย่าง ซึ่งเป็นค่ากึ่งกลางของค่าสังเกตทั้งหมด โดยหาได้จากผลรวมของค่าสังเกตทั้งหมด หารด้วยขนาดตัวอย่าง (ฤดี มาสุจันท์, 2554)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ n คือ จำนวนข้อมูลทั้งหมด หรือขนาดตัวอย่าง

x_i คือ ค่าสังเกตที่ i

ค่าเฉลี่ยตัวอย่าง สามารถหาได้ดังแสดงในสมการที่ 2.10

$$\bar{x} = \frac{x_1+x_2+x_3+..+x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n} \quad (2.10)$$

2.5 การคำนวณค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง (Sample Standard Deviation, s)

ส่วนเบี่ยงเบนมาตรฐานตัวอย่าง เป็นค่าที่บ่งบอกถึงการกระจายของข้อมูลในตัวอย่าง สามารถหาได้จากรากที่สองของความแปรปรวนตัวอย่าง ดังนั้น หน่วยของส่วนเบี่ยงเบนมาตรฐานตัวอย่าง จึงเป็นหน่วยเดียวกันกับหน่วยของข้อมูล (ฤดี มาสุจันท์, 2554)

กำหนดให้ n คือ จำนวนข้อมูลทั้งหมด หรือขนาดตัวอย่าง

x_i คือ ค่าสังเกตที่ i

ส่วนเบี่ยงเบนมาตรฐานตัวอย่าง สามารถหาได้ดังแสดงในสมการที่ 2.11

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \quad (2.11)$$

2.6 การประมาณค่า

การประมาณค่า คือ การประมาณค่าพารามิเตอร์ หรือลักษณะของประชากรโดยใช้ข้อมูลจากตัวอย่าง เป็นการประมาณค่าพารามิเตอร์ของประชากร โดยใช้ตัวสถิติที่เหมาะสม

การประมาณค่าแบบช่วง (Interval Estimation) เป็นการประมาณค่าพารามิเตอร์ของประชากร ว่าจะอยู่ในช่วงใดช่วงหนึ่ง โดยใช้ข้อมูลที่เก็บรวบรวมได้จากตัวอย่าง โดยที่ช่วงของการประมาณค่า จะบอกถึงค่าที่ต่ำสุดและค่าที่สูงสุดของพารามิเตอร์ เช่น ถ้าประมาณได้ว่า $\mathrm{L} < \mu < \mathrm{U}$ หมายความว่า ค่าเฉลี่ยประชากร (μ) มีค่าอยู่ระหว่าง L และ U

ระดับความเชื่อมั่น (Confidence Level) หมายถึง โอกาสที่พารามิเตอร์ของประชากร จะอยู่ในช่วงของค่าที่ประมาณได้ เช่น $P(L < \mu < U) = 0.95$ หมายถึง โอกาสที่ค่า μ จะมีค่าอยู่ในช่วง L และ U เป็น 0.95 หรือ 95 %

หรือกล่าวได้ว่า ในการสุ่มตัวอย่างขนาด n จากประชากร 100 ครั้ง ค่าเฉลี่ยประชากร μ จะมีค่าอยู่ในช่วง L และ U 95 ครั้ง จะมีเพียง 5 ครั้ง ที่ค่า μ ไม่อยู่ในช่วง L และ U จึงกล่าวได้ว่า μ มีค่าอยู่ในช่วง L และ U ด้วยระดับความเชื่อมั่น 95 %

ในกรณีทั่วไป จะให้ระดับความเชื่อมั่นเป็น $(1-\alpha)100\%$ หมายความว่า โอกาสที่การประมาณค่า จะผิดพลาด เป็น $\alpha 100\%$ ถ้าให้ระดับความเชื่อมั่น $1-\alpha$ มีค่ามาก จะทำให้โอกาสของความผิดพลาดในการประมาณค่า α มีค่าน้อย โอกาสของความผิดพลาด เรียกอีกอย่างหนึ่งว่า ระดับนัยสำคัญ (Significance level) ดังนั้น $P(L < \mu < U) = 1-\alpha$ (สายชล สินสมบูรณ์ทอง, 2553)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 ทฤษฎีที่เกี่ยวกับการประมาณค่าเฉลี่ยประชากรเดียวแบบช่วง

กรณีที่ประชากรมีการแจกแจงแบบใดๆ ไม่ทราบค่าความแปรปรวนของประชากร และตัวอย่างมีขนาดใหญ่ (ไม่ทราบค่า σ^2และ $n \geq 30$)

จากทฤษฎีขีดจำกัดสู่ส่วนกลาง (The Central Limit Theorem) ที่กล่าวว่า ถ้าสุ่มตัวอย่างขนาดใหญ่ $(n \geq 30)$ จากประชากรที่มีการแจกแจงแบบใดๆ ที่ไม่ใช่แบบปกติ และไม่ทราบค่าความแปรปรวน ของประชากร จะได้ว่าค่าเฉลี่ยตัวอย่าง $\bar{x}$ จะมีการแจกแจงโดยประมาณแบบปกติ โดยมีค่าเฉลี่ย $\mu_{\bar{x}} = \mu$ และความแปรปรวน $\sigma^2{}_{\bar{x}} = \frac{\sigma^2}{n}$

เมื่อไม่ทราบค่าความแปรปรวนของประชากร σ^2 และตัวอย่างที่สุ่มมาจากประชากรมีขนาดใหญ่ จะใช้ความแปรปรวนของตัวอย่าง s^2 แทนค่าความแปรปรวนของประชากร σ^2 ดังแสดงในสมการที่ 2.12 (สายชล สินสมบูรณ์ทอง, 2553)

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \tag{2.12}$$

ถ้าแปลงตัวแปร $\bar{x}$ ให้เป็นตัวแปร z จะได้ดังแสดงในสมการที่ 2.13

$$z = \frac{\bar{x} - \mu_{\bar{x}}}{\sigma_{\bar{x}}} = \frac{\bar{x} - \mu}{s/\sqrt{n}} \sim \mathrm{N}(0,1) \tag{2.13}$$

เมื่อกำหนดระดับความเชื่อมั่นให้เท่ากับ $1-\alpha$ ดังนั้น ช่วงความเชื่อมั่น $(1-\alpha)100\%$ ของ μ แสดงในสมการที่ 2.14

$$\bar{x} - \mathrm{z}_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}} < \quad \mu \quad < \bar{x} + \mathrm{z}_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}} \tag{2.14}$$

หรือ $\left(\bar{x} - \mathrm{z}_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}}\,,\, \bar{x} + \mathrm{z}_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}}\right)$

หรือ $\bar{x} \pm \mathrm{z}_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}}$

2.6.2 ทฤษฎีที่เกี่ยวกับการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง

กรณีที่ประชากรมีการแจกแจงแบบใดๆ ไม่ทราบค่าความแปรปรวนของประชากร และตัวอย่างมีขนาดใหญ่ (ไม่ทราบค่า σ_1^2 , σ_2^2และ $n_1 \geq 30$, $n_2 \geq 30$)

จากทฤษฎีขีดจำกัดสู่ส่วนกลางที่กล่าวว่า ถ้าสุ่มตัวอย่างขนาดใหญ่ $(n_1 \geq 30\,,\, n_2 \geq 30)$ จากประชากรที่มีการแจกแจงแบบใดๆ ที่ไม่ใช่แบบปกติ และไม่ทราบค่าความแปรปรวน ของประชากร จะได้ว่า $\bar{x}_1 - \bar{x}_2$ จะมีการแจกแจงโดยประมาณแบบปกติ โดยมีค่าเฉลี่ย $\mu_1 - \mu_2$ และความแปรปรวน $\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}$

เมื่อไม่ทราบค่าความแปรปรวนของประชากร σ_1^2 และ σ_2^2 จะประมาณค่าด้วยความแปรปรวนของตัวอย่าง s_1^2 และ s_2^2 นั่นคือ ประมาณ σ_1^2ด้วย s_1^2 และประมาณ σ_2^2 และ s_2^2

จะได้ว่า $\bar{x}_1 - \bar{x}_2 \sim N(\mu_1 - \mu_2\,,\, \frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าแปลงตัวแปร $\bar{x}$ ให้เป็นตัวแปร z จะได้ดังแสดงในสมการที่ 2.15 (สายชล สินสมบูรณ์ทอง, 2553)

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \sim \mathrm{N}(0,1) \qquad (2.15)$$

เมื่อกำหนดระดับความเชื่อมั่นให้เท่ากับ $1-\alpha$ ดังนั้น ช่วงความเชื่อมั่น $(1-\alpha)100\%$ ของ $\mu_1 - \mu_2$ แสดงในสมการที่ 2.16

$$(\bar{x}_1 - \bar{x}_2) - \mathrm{z}_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} < \quad \mu_1 - \mu_2 \quad < (\bar{x}_1 - \bar{x}_2) + \mathrm{z}_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \qquad (2.16)$$

หรือ $$((\bar{x}_1 - \bar{x}_2) - \mathrm{z}_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}\,, (\bar{x}_1 - \bar{x}_2) + \mathrm{z}_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}})$$

หรือ $$(\bar{x}_1 - \bar{x}_2) \pm \mathrm{z}_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

การพัฒนาโปรแกรมคอมพิวเตอร์ เพื่อใช้ในการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้กระบวนการตัดสินใจมาร์คอฟ ได้มีการศึกษาถึงทฤษฎีและปัจจัยที่ต้องใช้ในการค้นหาเส้นทางเดินของหุ่นยนต์ และมีการนำโปรแกรมคอมพิวเตอร์เข้ามาประยุกต์ใช้ เพื่อช่วยในการค้นหาเส้นทางเดินของหุ่นยนต์ โดยมีวิธีการดำเนินงานและหัวข้อที่จะศึกษาดังนี้

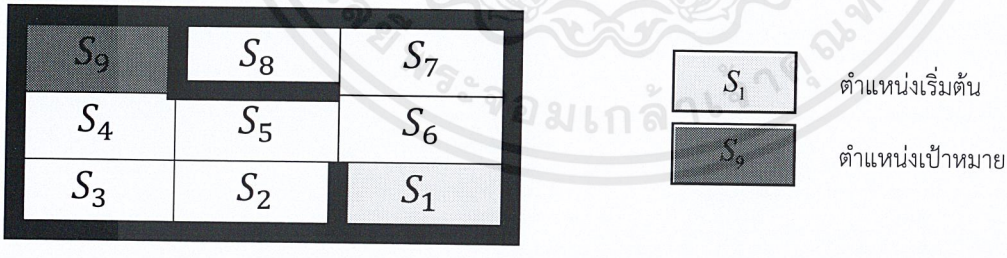
1. การศึกษาความเป็นไปได้ของการนำทฤษฎีของมาร์คอฟ มาใช้ค้นหาเส้นทางเดินของหุ่นยนต์
2. การศึกษาปัจจัยที่มีผลต่อการปรับเปลี่ยนสถานะจากตำแหน่งปัจจุบัน ไปยังตำแหน่งถัดไป
3. การจัดทำโปรแกรมคอมพิวเตอร์สำหรับจำลองการค้นหาเส้นทางเดินของหุ่นยนต์

3.1 การศึกษาความเป็นไปได้ของการนำทฤษฎีของมาร์คอฟ เพื่อใช้ค้นหาเส้นทางเดินของหุ่นยนต์

ในการศึกษาได้ใช้ตารางขนาด 3x3 มาทำการทดลอง เพื่อหาความเป็นไปได้ที่จะนำทฤษฎีของมาร์คอฟมาใช้หาเส้นทางเดินของหุ่นยนต์ ซึ่งพิจารณาจากค่าความน่าจะเป็นของการเปลี่ยนสถานะเริ่มเข้าสู่สภาวะคงที่ โดยใช้เมตริกซ์เดิมคูณตัวเองเป็นจำนวนครั้ง n โดย n คือจำนวนครั้งที่ทำให้เข้าสู่สถานะคงที่

<u>กรณีที่ 1</u> เป็นการกำหนดตำแหน่งสิ่งกีดขวาง ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

การทดลองเริ่มจากการสร้างตารางขนาด 3x3 ขึ้นมา โดยมีการกำหนดชื่อของแต่ละตำแหน่ง และมีการกำหนดตำแหน่งของจุดเริ่มต้น (S_1) และตำแหน่งของเป้าหมาย (S_9) นอกจากนั้นยังมีการกำหนดสิ่งกีดขวาง ดังรูปที่ 3.1



รูปที่ 3.1 แสดงตำแหน่งจุดเริ่มต้น เป้าหมาย และสิ่งกีดขวางมี โดยมีช่องทางที่สามารถไปยังเป้าหมายได้

สร้างเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ $P_{ij}^{n}; i = S_1, S_2 \ldots., S_9\ ,\ j = S_1, S_2 \ldots., S_9$ โดยมีการให้ความน่าจะเป็นในการที่จะไปในแต่ละทิศทางเท่าๆกัน แต่ด้านที่เป็นขอบของตารางและสิ่งกีดขวางจะกำหนดให้มีความน่าจะเป็นเท่ากับศูนย์ ซึ่งจะได้ตารางเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ ขนาด 9x9 ดังตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของการคำนวณตัวเลขในตารางที่ 3.1 เช่น ถ้าพิจารณาที่สถานะที่ 1 จากรูปที่ 3.1 จะเห็นได้ว่ามีสิ่งกีดขวางด้านซ้ายมือ ส่วนด้านขวาและด้านล่างเป็นขอบของตาราง ดังนั้นจึงเหลือทิศทางที่จะสามารถไปได้มี 2 ทิศทาง คือ ด้านบนและอยู่ตำแหน่งเดิม ดังนั้นความน่าจะเป็นของการเปลี่ยนจากสถานะ S_1 ไปยัง S_1 และความน่าจะเป็นของการเปลี่ยนจากสถานะ S_1 ไปยัง S_9 เท่ากับ 0.5

ตารางที่ 3.1 แสดงเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^1 เมื่อกำหนดตำแหน่งสิ่งกีดขวางให้มีช่องทางที่สามารถไปยังเป้าหมายได้

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	1/2	0	0	0	0	1/2	0	0	0
S_2	0	1/3	1/3	0	1/3	0	0	0	0
S_3	0	1/3	1/3	1/3	0	0	0	0	0
S_4	0	0	1/4	1/4	1/4	0	0	0	1/4
S_5	0	1/4	0	1/4	1/4	1/4	0	0	0
S_6	1/4	0	0	0	1/4	1/4	1/4	0	0
S_7	0	0	0	0	0	1/3	1/3	1/3	0
S_8	0	0	0	0	0	0	1/2	1/2	0
S_9	0	0	0	1/2	0	0	0	0	1/2

จากนั้นทำการคำนวณโดยใช้เมตริกซ์เดิมคูณตัวเอง ซึ่งในที่นี้เราจะทำการคูณเมตริกซ์เดิมด้วยตัวเองเป็นจำนวนครั้ง n โดย n คือจำนวนครั้งที่ทำให้เข้าสู่สถานะคงที่

ในการทดลองได้ทำการคูณเมตริกซ์ในโปรแกรม Matlab ซึ่งได้ผลการคำนวณ ดังตารางที่ 3.2 (a), 3.2 (b) และ 3.2 (c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 (a) แสดงเมตริกซ์เริ่มต้นในการคำนวณในโปรแกรม Matlab เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

A =	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	1/2	0	0	0	0	1/2	0	0	0
S_2	0	1/3	1/3	0	1/3	0	0	0	0
S_3	0	1/3	1/3	1/3	0	0	0	0	0
S_4	0	0	1/4	1/4	1/4	0	0	0	1/4
S_5	0	1/4	0	1/4	1/4	1/4	0	0	0
S_6	1/4	0	0	0	1/4	1/4	1/4	0	0
S_7	0	0	0	0	0	1/3	1/3	1/3	0
S_8	0	0	0	0	0	0	1/2	1/2	0
S_9	0	0	0	1/2	0	0	0	0	1/2

ตารางที่ 3.2 (b) แสดงผลการคำนวณจากโปรแกรม Matlab โดยเมตริกซ์เดิมคูณตัวเองเป็นจำนวน 2 ครั้ง เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

A^2=	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.375	0	0	0	0.125	0.375	0.125	0	0
S_2	0	0.3056	0.2222	0.1944	0.1944	0.0833	0	0	0
S_3	0	0.2222	0.3056	0.1944	0.1944	0	0	0	0.0833
S_4	0	0.1458	0.1458	0.3333	0.125	0.0625	0	0	0.1875
S_5	0.0625	0.1458	0.1458	0.125	0.2708	0.125	0.0625	0	0.0625
S_6	0.1875	0.0625	0	0.0625	0.125	0.3333	0.1458	0.0833	0
S_7	0.0833	0	0	0	0.0833	0.1944	0.3611	0.2778	0
S_8	0	0	0	0	0	0.1667	0.4167	0.4167	0
S_9	0	0	0.125	0.375	0.125	0	0	0	0.375

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 (c) แสดงผลการคำนวณจากโปรแกรม Matlab โดยเมตริกซ์เดิมคูณตัวเองเป็นจำนวน 150 ครั้ง เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

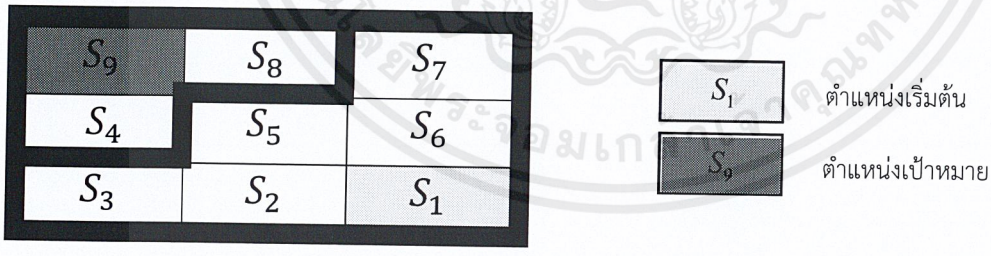
A^{150} =

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_2	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_3	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_4	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_5	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_6	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_7	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_8	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741
S_9	0.0741	0.1111	0.1111	0.1481	0.1481	0.1481	0.1111	0.0741	0.0741

จากผลการคำนวณ n ในที่นี้คือ 150 และความน่าจะเป็นของการเปลี่ยนสถานะP_{ij}^{n} ; $= S_1$, $j = S_9$ เท่ากับ 0.0741 > 0 ซึ่งแสดงให้เห็นว่า มีความน่าจะเป็นที่จะไปยังเป้าหมายได้

<u>กรณีที่ 2</u> เป็นการกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

โดยให้เคลื่อนที่จากตำแหน่งเริ่มต้น (S_1) ไปยังตำแหน่งเป้าหมาย (S_9) และสร้างตารางขนาด 3x3 ขึ้น โดยมีการกำหนดชื่อของแต่ละตำแหน่ง และมีการกำหนดตำแหน่งของจุดเริ่มต้น (S_1) และตำแหน่งของเป้าหมาย (S_9) นอกจากนั้นยังมีการกำหนดสิ่งกีดขวาง ดังรูปที่ 3.2



รูปที่ 3.2 แสดงตำแหน่งจุดเริ่มต้น เป้าหมาย และสิ่งกีดขวาง โดยไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

สร้างเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^{n} ; $i = S_1, S_2 \ldots, S_9$, $j = S_1, S_2 \ldots, S_9$ โดยมีการให้ความน่าจะเป็นในการที่จะไปในแต่ละทิศทางเท่าๆกัน แต่ด้านที่เป็นขอบของตารางและสิ่งกีดขวางจะกำหนดให้มีความน่าจะเป็นเท่ากับศูนย์ ซึ่งจะได้ตารางเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ ขนาด 9x9 ดังตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 แสดงเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^{1} เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	1/3	1/3	0	0	0	1/3	0	0	0
S_2	1/4	1/4	1/4	0	1/4	0	0	0	0
S_3	0	1/2	1/2	0	0	0	0	0	0
S_4	0	0	0	1/2	0	0	0	0	1/2
S_5	0	1/3	0	0	1/3	1/3	0	0	0
S_6	1/4	0	0	0	1/4	1/4	1/4	0	0
S_7	0	0	0	0	0	1/2	1/2	0	0
S_8	0	0	0	0	0	0	0	1/2	1/2
S_9	0	0	0	1/3	0	0	0	1/3	1/3

จากนั้นทำการคำนวณโดยใช้เมตริกซ์เดิมคูณตัวเอง ซึ่งในที่นี้เราจะทำการคูณเมตริกซ์เดิมด้วยตัวเองเป็นจำนวนครั้ง n โดย n คือจำนวนครั้งที่ทำให้เข้าสู่สถานะคงที่

ในการทดลองได้ทำการคูณเมตริกซ์ในโปรแกรม Matlab ซึ่งได้ผลการคำนวณ ดังตารางที่ 3.4 (a), 3.4 (b) และ 3.4 (c)

ตารางที่ 3.4 (a) แสดงเมตริกซ์เริ่มต้นในการคำนวณในโปรแกรม Matlab เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

B =	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	1/3	1/3	0	0	0	1/3	0	0	0
S_2	1/4	1/4	1/4	0	1/4	0	0	0	0
S_3	0	1/2	1/2	0	0	0	0	0	0
S_4	0	0	0	1/2	0	0	0	0	1/2
S_5	0	1/3	0	0	1/3	1/3	0	0	0
S_6	1/4	0	0	0	1/4	1/4	1/4	0	0
S_7	0	0	0	0	0	1/2	1/2	0	0
S_8	0	0	0	0	0	0	0	1/2	1/2
S_9	0	0	0	1/3	0	0	0	1/3	1/3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 (b) แสดงผลการคำนวณจากโปรแกรม Matlab โดยเมตริกซ์เดิมคูณตัวเองเป็นจำนวน 2 ครั้ง เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

B^2 =

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.2778	0.1944	0.0833	0	0.1667	0.1944	0.0833	0	0
S_2	0.1458	0.3542	0.1875	0	0.1458	0.1667	0	0	0
S_3	0.125	0.375	0.375	0	0.125	0	0	0	0
S_4	0	0	0	0.4167	0	0	0	0.1667	0.4167
S_5	0.1667	0.1944	0.0833	0	0.2778	0.1944	0.0833	0	0
S_6	0.1458	0.1667	0	0	0.1458	0.3542	0.1875	0	0
S_7	0.125	0	0	0	0.125	0.375	0.375	0	0
S_8	0	0	0	0.1667	0	0	0	0.4167	0.4167
S_9	0	0	0	0.2778	0	0	0	0.2778	0.4444

ตารางที่ 3.4 (c) แสดงผลการคำนวณจากโปรแกรม Matlab โดยเมตริกซ์เดิมคูณตัวเองเป็นจำนวน 150 ครั้ง เมื่อกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้

B^{150} =

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.1667	0.2222	0.1111	0	0.1667	0.2222	0.1111	0	0
S_2	0.1667	0.2222	0.1111	0	0.1667	0.2222	0.1111	0	0
S_3	0.1667	0.2222	0.1111	0	0.1667	0.2222	0.1111	0	0
S_4	0	0	0	0.2857	0	0	0	0.2857	0.4286
S_5	0.1667	0.2222	0.1111	0	0.1667	0.2222	0.1111	0	0
S_6	0.1667	0.2222	0.1111	0	0.1667	0.2222	0.1111	0	0
S_7	0.1667	0.2222	0.1111	0	0.1667	0.2222	0.1111	0	0
S_8	0	0	0	0.2857	0	0	0	0.2857	0.4286
S_9	0	0	0	0.2857	0	0	0	0.2857	0.4286

จากผลการคำนวณ n ในที่นี้คือ 150 และความน่าจะเป็นของการเปลี่ยนสถานะP_{ij}^{n} ; $i = S_1$, $j = S_9$ เท่ากับ 0 ซึ่งแสดงให้เห็นว่า ไม่มีความน่าจะเป็นที่จะไปยังเป้าหมายจาก S_1ถึง S_9

ดังนั้นจากการทดลองทั้ง 2 กรณี จึงสรุปได้ว่า มีความเป็นไปได้ที่จะนำทฤษฎีของมาร์คอฟ มาประยุกต์ใช้ในการค้นหาเส้นทางเดินของหุ่นยนต์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การศึกษาปัจจัยที่มีผลต่อการปรับเปลี่ยนสถานะ

ในการศึกษาจะมีการใช้ตารางขนาด 3x3 มาทำการทดลอง เพื่อศึกษาปัจจัยที่มีผลต่อการปรับเปลี่ยนสถานะ โดยเริ่มจากการสร้างตารางขนาด 3x3 ขึ้นมา ดังรูปที่ 3.1

จากนั้นสร้างเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^{n} ; $i = S_1, S_2 \ldots., S_9$, $j = S_1, S_2 \ldots., S_9$ โดยมีการให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า และด้านที่เป็นขอบของตารางและสิ่งกีดขวางจะกำหนดให้มีความน่าจะเป็นเท่ากับศูนย์ ซึ่งจะได้ตารางเมตริกซ์ขนาดความน่าจะเป็นของการเปลี่ยนสถานะ 9x9 ดังตารางที่ 3.5

ตารางที่ 3.5 แสดงเมตริกซ์ความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^{1} เมื่อให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.4	0	0	0	0	0.6	0	0	0
S_2	0	0.2	0.6	0	0.2	0	0	0	0
S_3	0	0.2	0.2	0.6	0	0	0	0	0
S_4	0	0	0.4/3	0.4/3	0.4/3	0	0	0	0.6
S_5	0	0.4/3	0	0.6	0.4/3	0.4/3	0	0	0
S_6	0.4/3	0	0	0	0.6	0.4/3	0.4/3	0	0
S_7	0	0	0	0	0	0.6	0.2	0.2	0
S_8	0	0	0	0	0	0	0.6	0.4	0
S_9	0	0	0	0.4	0	0	0	0	0.6

จากนั้นทำการคำนวณโดยใช้เมตริกซ์เดิมคูณตัวเอง ซึ่งในที่นี้เราจะทำการคูณเมตริกซ์เดิมด้วยตัวเองเป็นจำนวนครั้ง n โดย n คือจำนวนครั้งที่ทำให้เข้าสู่สถานะคงที่

ในการทดลองได้ทำการคูณเมตริกซ์ในโปรแกรม Matlab ซึ่งได้ผลการคำนวณ ดังตารางที่ 3.6 (a), 3.6 (b) และ 3.7 (c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 (a) แสดงเมตริกซ์เริ่มต้นในการคำนวณในโปรแกรม Matlab เมื่อให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า

C =	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.4	0	0	0	0	0.6	0	0	0
S_2	0	0.2	0.6	0	0.2	0	0	0	0
S_3	0	0.2	0.2	0.6	0	0	0	0	0
S_4	0	0	0.4/3	0.4/3	0.4/3	0	0	0	0.6
S_5	0	0.4/3	0	0.6	0.4/3	0.4/3	0	0	0
S_6	0.4/3	0	0	0	0.6	0.4/3	0.4/3	0	0
S_7	0	0	0	0	0	0.6	0.2	0.2	0
S_8	0	0	0	0	0	0	0.6	0.4	0
S_9	0	0	0	0.4	0	0	0	0	0.6

ตารางที่ 3.6 (b) แสดงผลการคำนวณจากโปรแกรม Matlab โดยเมตริกซ์เดิมคูณตัวเองเป็นจำนวน 2 ครั้ง เมื่อให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า

C^2 =	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.24	0	0	0	0.36	0.32	0.08	0	0
S_2	0	0.1867	0.24	0.48	0.0667	0.0267	0	0	0
S_3	0	0.08	0.24	0.2	0.12	0	0	0	0.36
S_4	0	0.0444	0.0444	0.4178	0.0356	0.0178	0	0	0.44
S_5	0.0178	0.0444	0.16	0.16	0.2044	0.0356	0.0178	0	0.36
S_6	0.0711	0.08	0	0.36	0.16	0.2578	0.0444	0.0267	0
S_7	0.08	0	0	0	0.36	0.2	0.24	0.12	0
S_8	0	0	0	0	0	0.36	0.36	0.28	0
S_9	0	0	0.0533	0.2933	0.0533	0	0	0	0.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 (c) แสดงผลการคำนวณจากโปรแกรม Matlab โดยเมตริกซ์เดิมคูณตัวเองเป็นจำนวน 150 ครั้ง เมื่อให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า

C^{150} =

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_2	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_3	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_4	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_5	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_6	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_7	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_8	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826
S_9	0.0033	0.0302	0.0763	0.3217	0.0667	0.0148	0.0033	0.0011	0.4826

จากผลการคำนวณ n ในที่นี้คือ 150 และความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^{150} ; $i = S_1$, $j = S_9$ เท่ากับ 0.4826

จากนั้นนำค่าความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^{150} ที่ได้จากการกำหนดให้ความน่าจะเป็นในการที่จะไปในแต่ละทิศทางเท่าๆกัน มาเปรียบเทียบกับค่าความน่าจะเป็นของการเปลี่ยนสถานะ P_{ij}^{150} ที่ได้จากการกำหนดให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า จะเห็นได้ว่าในการที่กำหนดให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า มีค่าความน่าจะเป็นของการเปลี่ยนสถานะจาก S_1 ไปยัง S_9 มากกว่า การกำหนดให้ความน่าจะเป็นในการที่จะไปในแต่ละทิศทางเท่าๆกัน (0.4826 > 0.0741)

ดังนั้นจึงสามารถสรุปได้ว่า การให้น้ำหนักมากหรือน้อยกับค่าความน่าจะเป็นของแต่ละทิศทาง มีผลต่อค่าความน่าจะเป็นของการปรับเปลี่ยนสถานะ

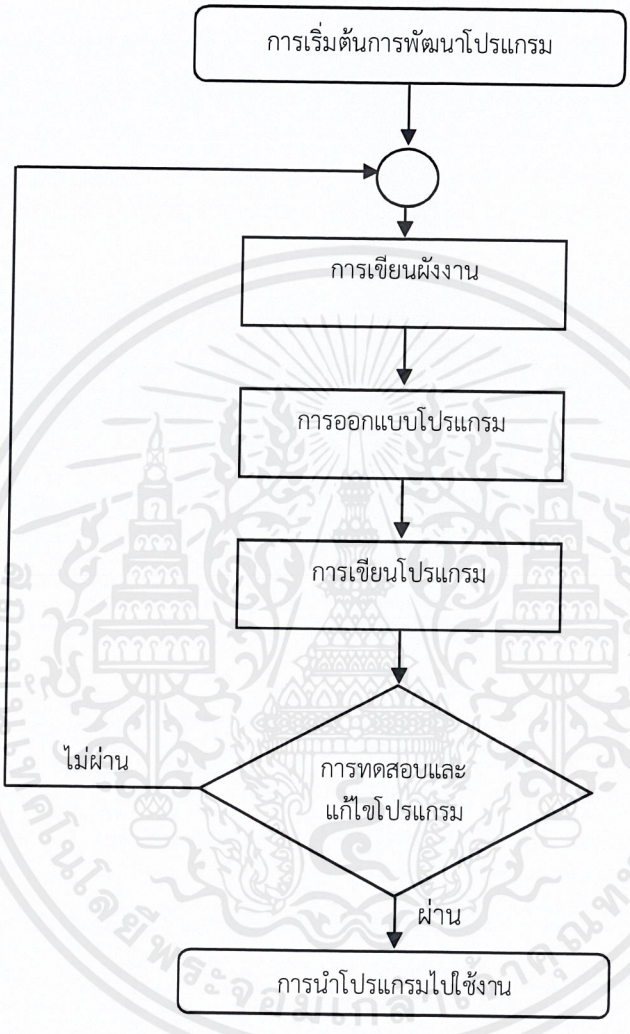
ทางกลุ่มจึงได้มีการกำหนดนโยบายขึ้นมา เพื่อใช้เป็นแนวทางในการปรับเปลี่ยนสถานะ โดยมีนโยบายต่างๆ ดังนี้

- กำหนดให้มีค่าความน่าจะเป็นในการที่จะไปในแต่ละทิศทางเท่าๆกัน
- กำหนดให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดทำโปรแกรมคอมพิวเตอร์สำหรับจำลองการค้นหาเส้นทางเดินของหุ่นยนต์

การจัดทำโปรแกรมคอมพิวเตอร์เพื่อใช้เป็นเครื่องมือจำลองการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้กระบวนการตัดสินใจมาร์คอฟ มีขั้นตอนการพัฒนาโปรแกรมที่สำคัญ 5 ขั้นตอน ดังนี้



รูปที่ 3.3 แสดงขั้นตอนการพัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

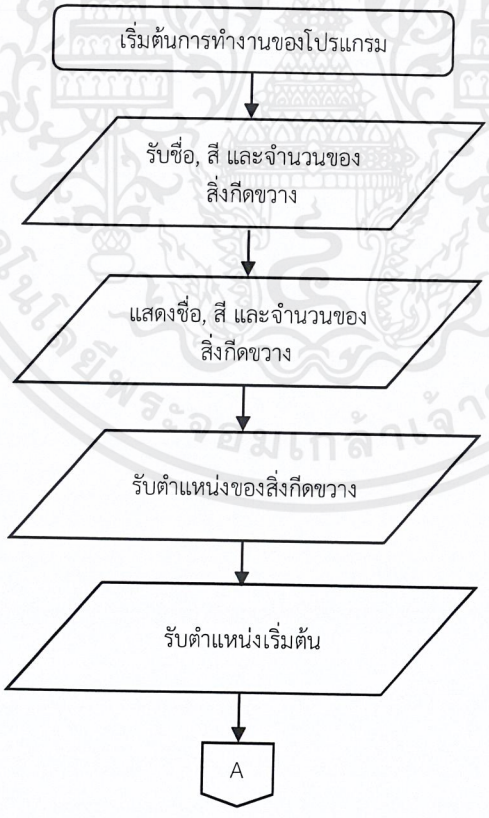
ขั้นตอนการพัฒนาโปรแกรมที่สำคัญ 5 ขั้นตอน ดังรูปที่ 3.3 สามารถอธิบายได้ดังนี้

3.3.1 การเขียนผังงาน

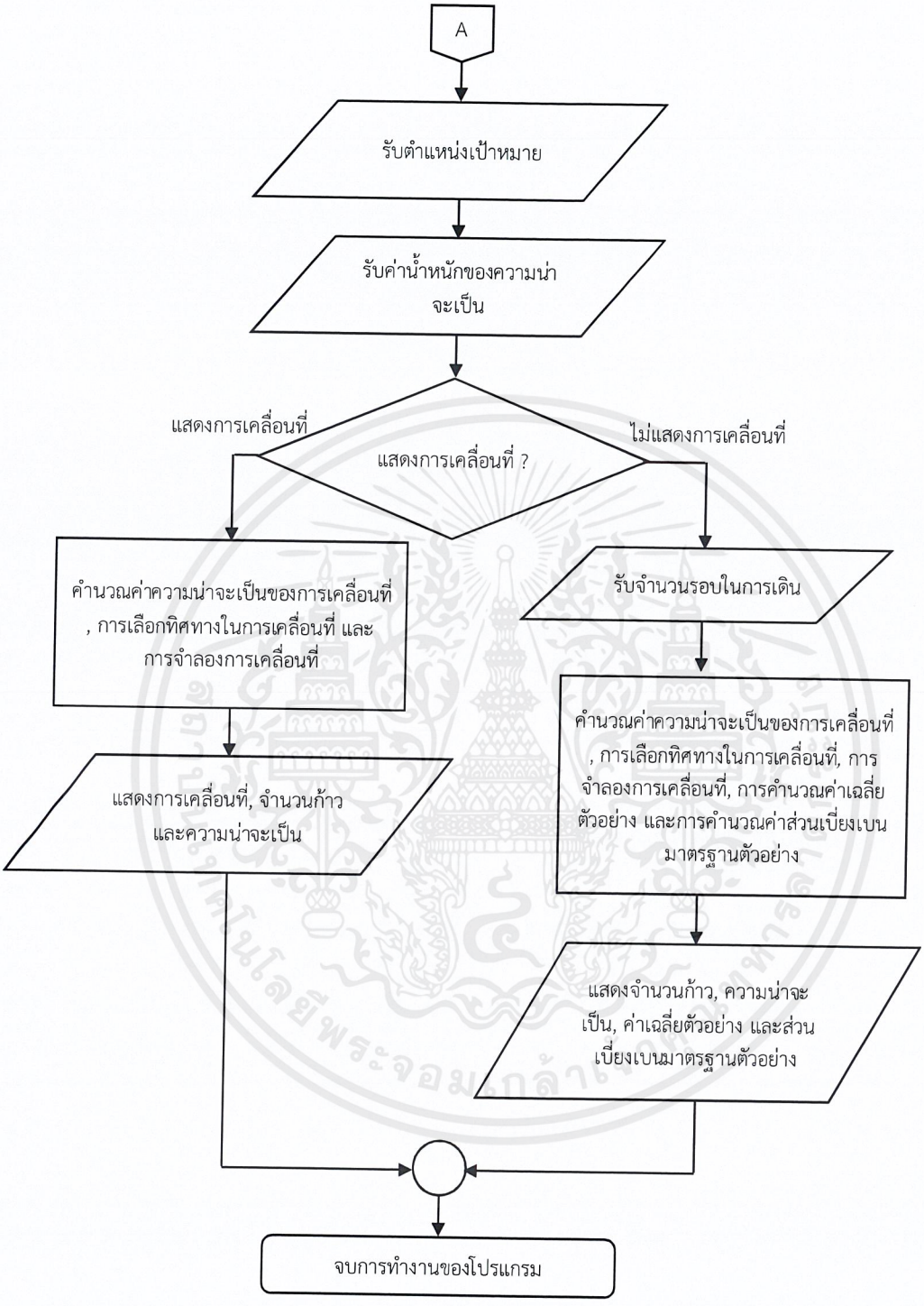
ผังงาน (Flow Chart) คือ แผนภาพแสดงลำดับขั้นตอนการทำงาน เป็นเครื่องมือที่ใช้ในการวางแผน โดยใช้สัญลักษณ์ต่างๆในการเขียนผังงาน เพื่อช่วยลำดับแนวความคิดในการเขียนโปรแกรม เป็นวิธีที่นิยมใช้เพราะทำให้เห็นภาพในการทำงานของโปรแกรม หากมีข้อผิดพลาด สามารถดูจากผังงาน จะทำให้การแก้ไขหรือปรับปรุงโปรแกรมทำได้ง่ายขึ้น

3.3.1.1 ผังงานแสดงการทำงานของโปรแกรมรวม

เมื่อทำการเปิดโปรแกรมคอมพิวเตอร์สำหรับจำลองการค้นหาเส้นทางเดินของหุ่นยนต์ ผู้ใช้จะต้องใส่ข้อมูลในหน้าต่างรับข้อมูลหรือฟอร์มที่ 1 และหน้าต่างในการแสดงผลจากการประมวลผลหรือฟอร์มที่ 2 ของโปรแกรม จากนั้นเลือกว่าต้องการที่จะแสดงการเคลื่อนที่หรือไม่ หากเลือกแสดงการเคลื่อนที่ (Show) ผู้ใช้จะสามารถกดปุ่ม Go เพื่อให้โปรแกรมทำการประมวลผล แต่ถ้าผู้ใช้เลือกไม่แสดงการเคลื่อนที่ (Not Show) ผู้ใช้จะต้องทำการใส่จำนวนรอบในการเดินก่อน ถึงจะสามารถกดปุ่ม Go เพื่อให้โปรแกรมทำการประมวลผล โดยผังงานแสดงการทำงานของโปรแกรมรวมแสดงดังรูปที่ 3.4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงผังงานการทำงานของโปรแกรมรวมรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 ผังงานแสดงการทำงานของโปรแกรมย่อย

จากรูปที่ 3.4 ในส่วนของการคำนวณต่างๆ สามารถแยกพิจารณาได้เป็นโปรแกรมย่อย ดังนี้

1. การคำนวณค่าความน่าจะเป็นของการเคลื่อนที่ สามารถอธิบายเป็นผังงาน ดังรูปที่ 3.5

โดยที่ wT คือ ค่าน้ำหนักของความน่าจะเป็นของด้านบน

wB คือ ค่าน้ำหนักของความน่าจะเป็นของด้านล่าง

wL คือ ค่าน้ำหนักของความน่าจะเป็นของด้านซ้าย

wR คือ ค่าน้ำหนักของความน่าจะเป็นของด้านขวา

ObT คือ สิ่งกีดขวางด้านบน; 0 = มี, 1 = ไม่มี

ObB คือ สิ่งกีดขวางด้านล่าง; 0 = มี, 1 = ไม่มี

ObL คือ สิ่งกีดขวางด้านซ้าย; 0 = มี, 1 = ไม่มี

ObR คือ สิ่งกีดขวางด้านขวา; 0 = มี, 1 = ไม่มี

SumProb คือ ผลรวมระหว่างผลคูณของน้ำหนักและความน่าจะเป็นของการเคลื่อนที่

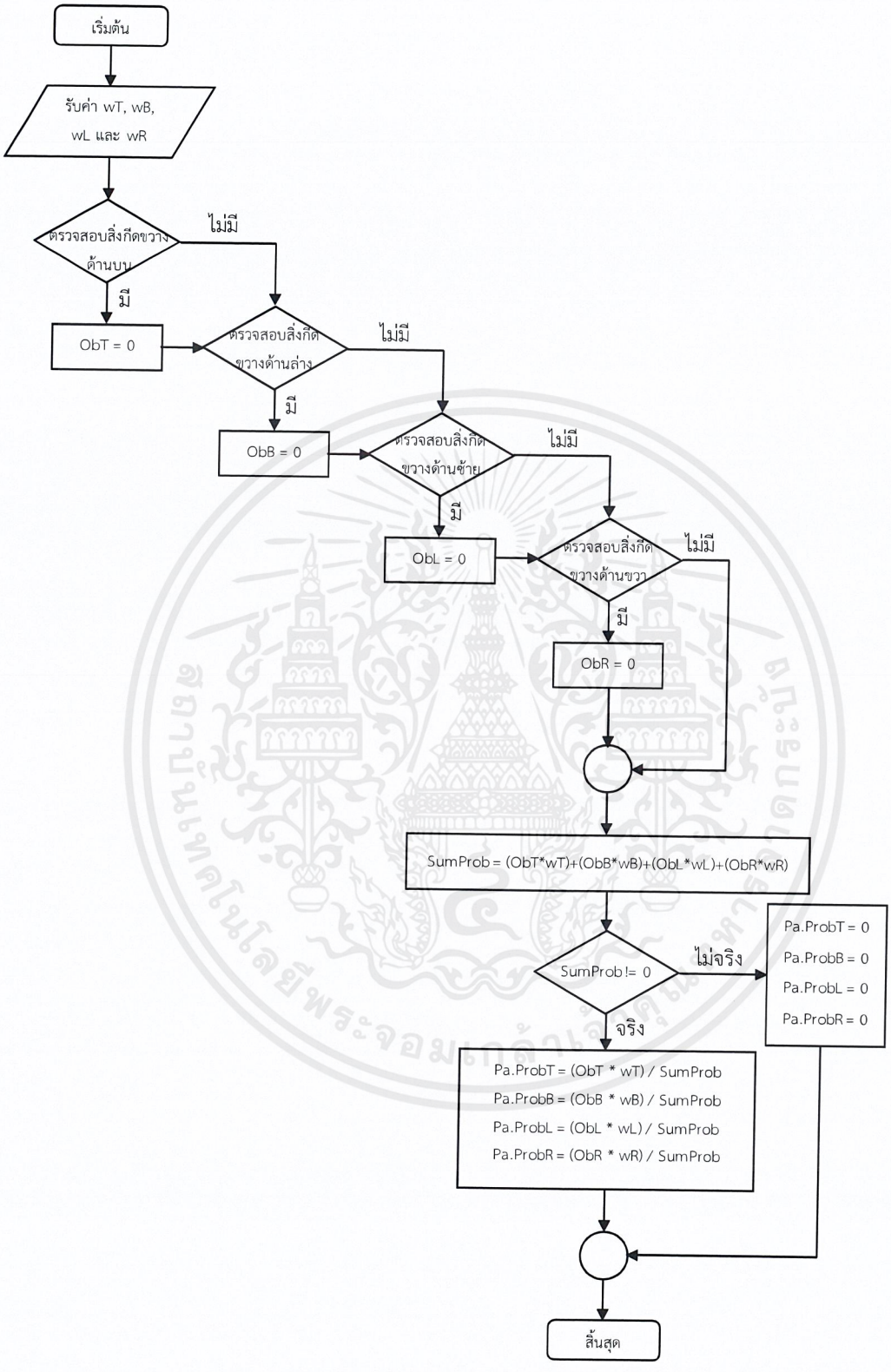
Pa.ProbT คือ ความน่าจะเป็นของการเคลื่อนที่ไปด้านบน

Pa.ProbB คือ ความน่าจะเป็นของการเคลื่อนที่ไปด้านล่าง

Pa.ProbL คือ ความน่าจะเป็นของการเคลื่อนที่ไปด้านซ้าย

Pa.ProbR คือ ความน่าจะเป็นของการเคลื่อนที่ไปด้านขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงผังงานการคำนวณค่าความน่าจะเป็นของการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

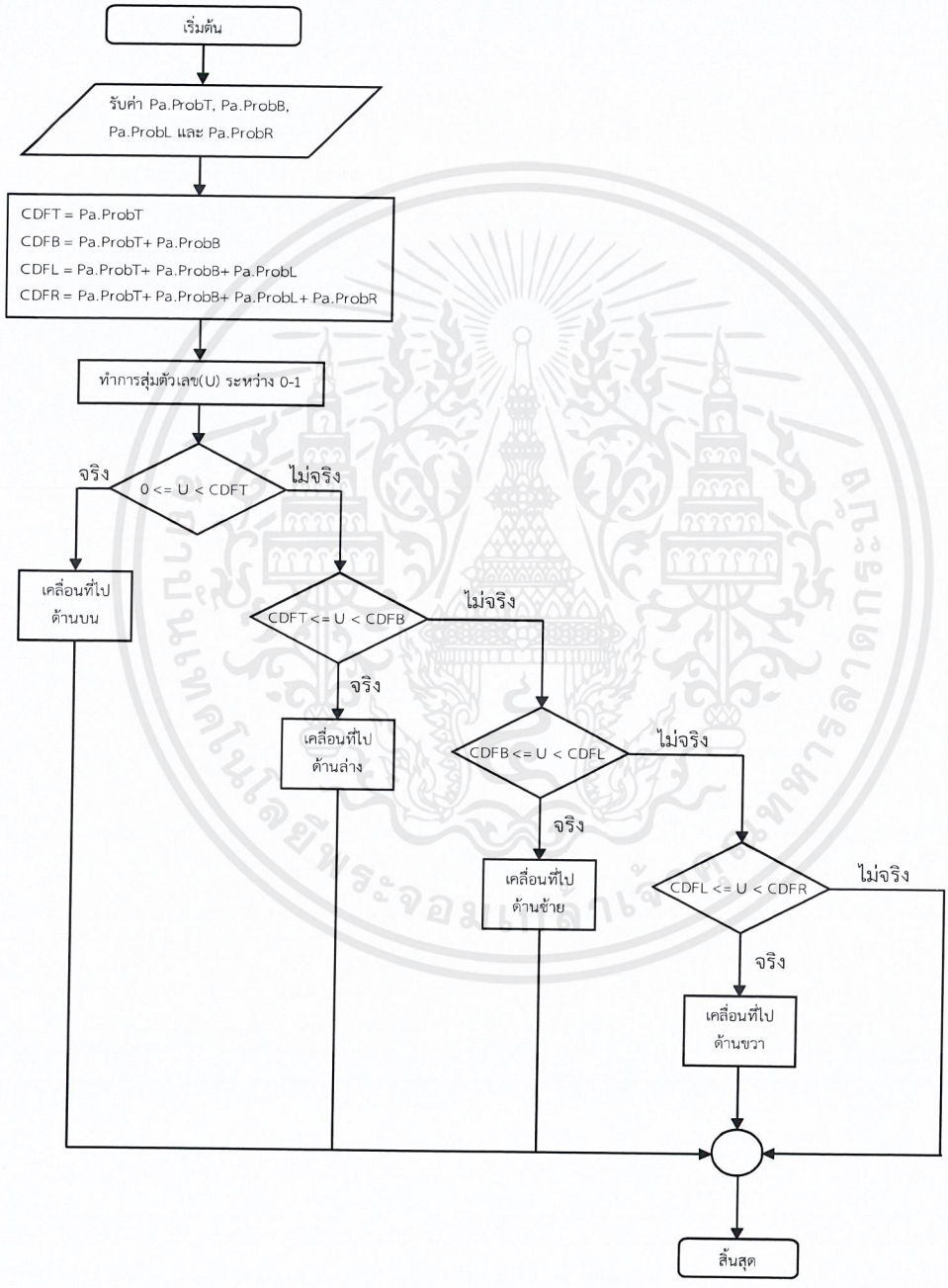
2. การเลือกทิศทางในการเคลื่อนที่ สามารถอธิบายเป็นผังงาน ดังรูปที่ 3.6

โดยที่ CDFT คือ ความน่าจะเป็นสะสมของด้านบน

CDFB คือ ความน่าจะเป็นสะสมของด้านบนและด้านล่าง

CDFL คือ ความน่าจะเป็นสะสมของด้านบน ด้านล่าง และด้านซ้าย

CDFR คือ ความน่าจะเป็นสะสมของด้านบน ด้านล่าง ด้านซ้าย และด้านขวา



รูปที่ 3.6แสดงผังงานการเลือกทิศทางในการเคลื่อนที่

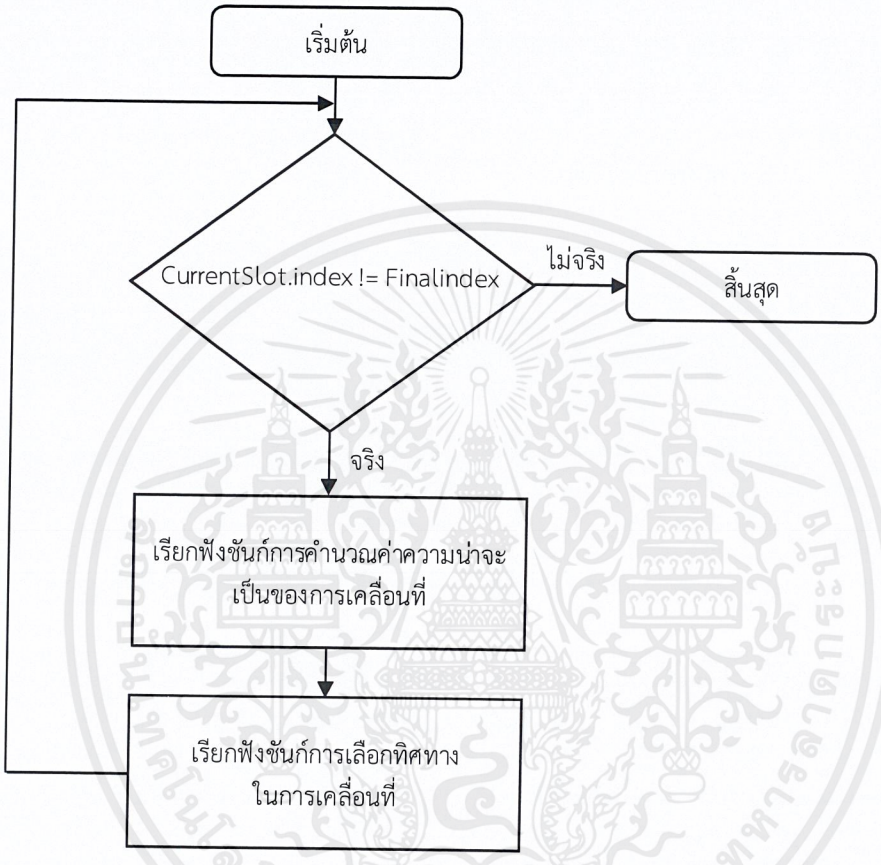
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การจำลองการเคลื่อนที่ สามารถอธิบายเป็นผังงาน ดังรูปที่ 3.7

โดยที่ CurrentSlot คือ ช่องปัจจุบัน

CurrentSlot.index คือ ตำแหน่งของช่องปัจจุบัน

Finalindex คือ ตำแหน่งของช่องเป้าหมาย



รูปที่ 3.7 แสดงการจำลองการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การคำนวณค่าเฉลี่ยตัวอย่าง สามารถอธิบายเป็นผังงาน ดังรูปที่ 3.8

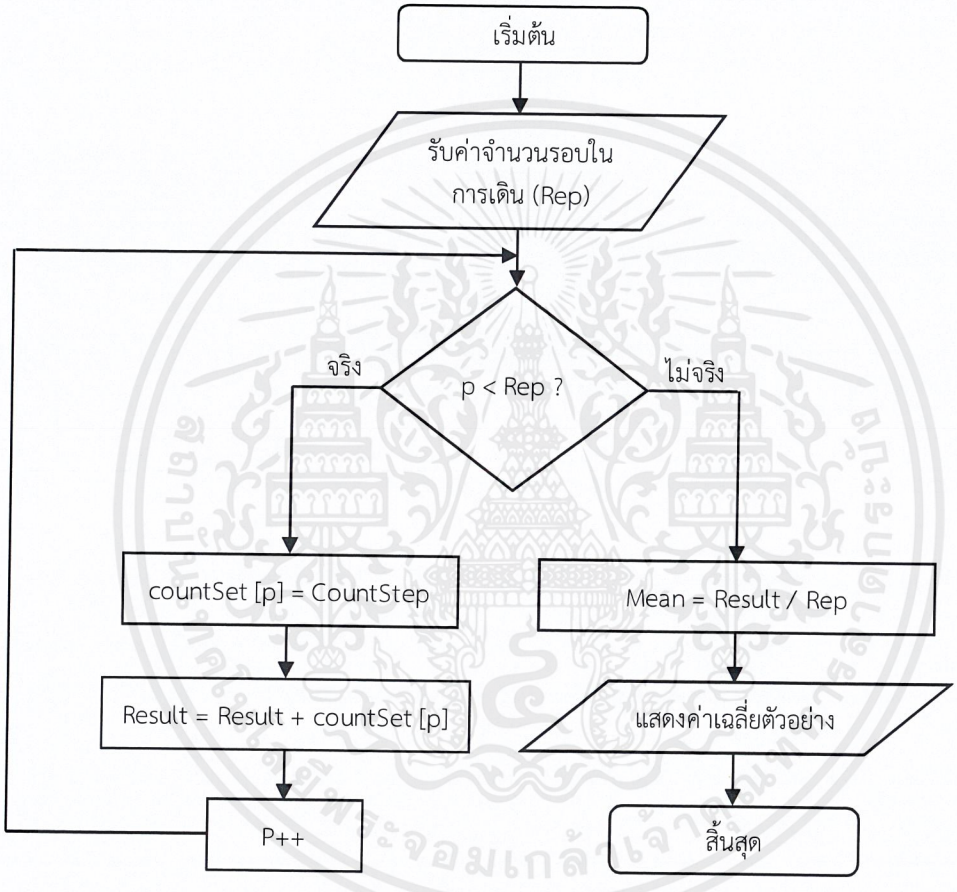
โดยที่ p คือ ตัวแปรที่สมมติขึ้น เพื่อตรวจสอบจำนวนรอบในการเดิน

Rep คือ จำนวนรอบในการเดิน

CountStep คือ นับจำนวนก้าว

Result คือ ผลรวมของจำนวนก้าว

Mean คือ ค่าเฉลี่ยตัวอย่าง



รูปที่ 3.8 แสดงการคำนวณค่าเฉลี่ยตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

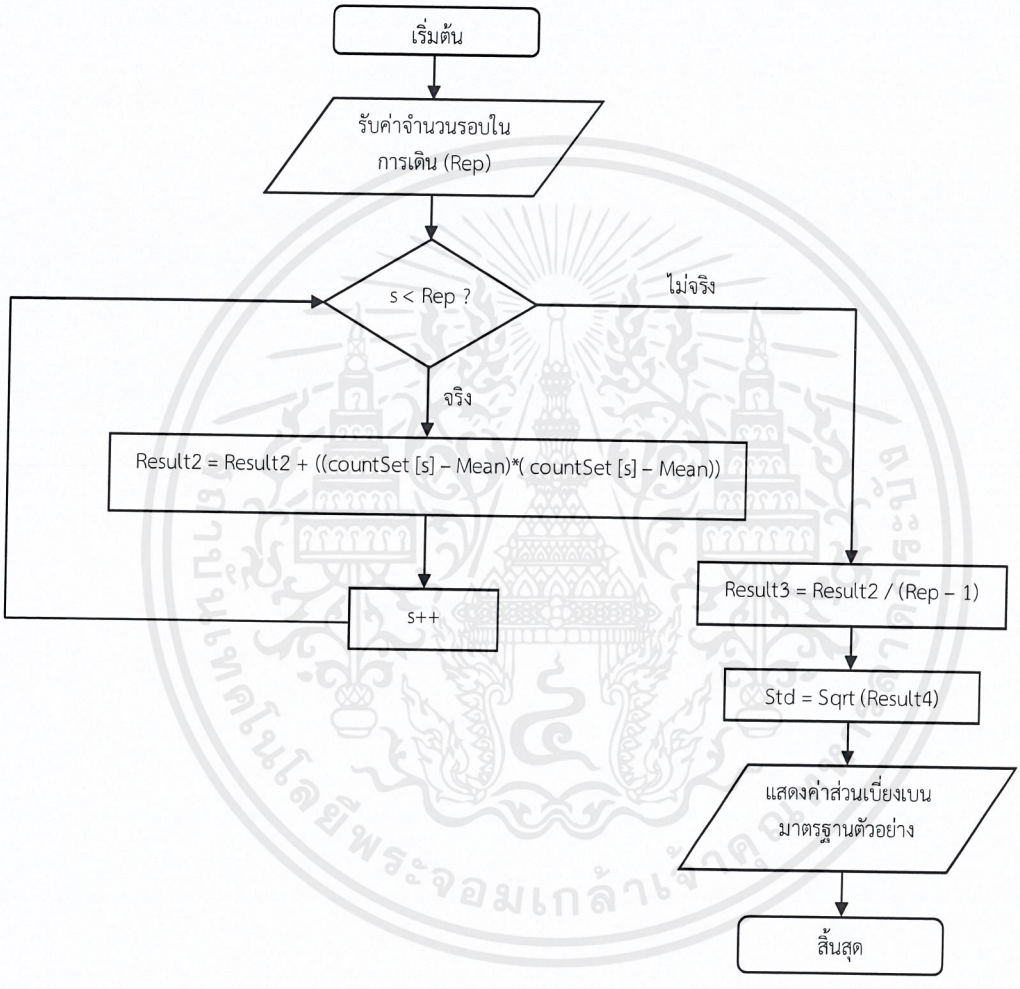
5. การคำนวณค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง สามารถอธิบายเป็นผังงาน ดังรูปที่ 3.9

โดยที่ s คือ ตัวแปรที่สมมติขึ้น เพื่อตรวจสอบจำนวนรอบในการเดิน

Result2 คือ $\sum(x_i - \bar{x})^2$

Result3 คือ $\frac{\sum(x_i-\bar{x})^2}{\mathbf{n-1}}$

Std คือ ค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง



รูปที่ 3.9 แสดงการคำนวณค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การออกแบบโปรแกรม

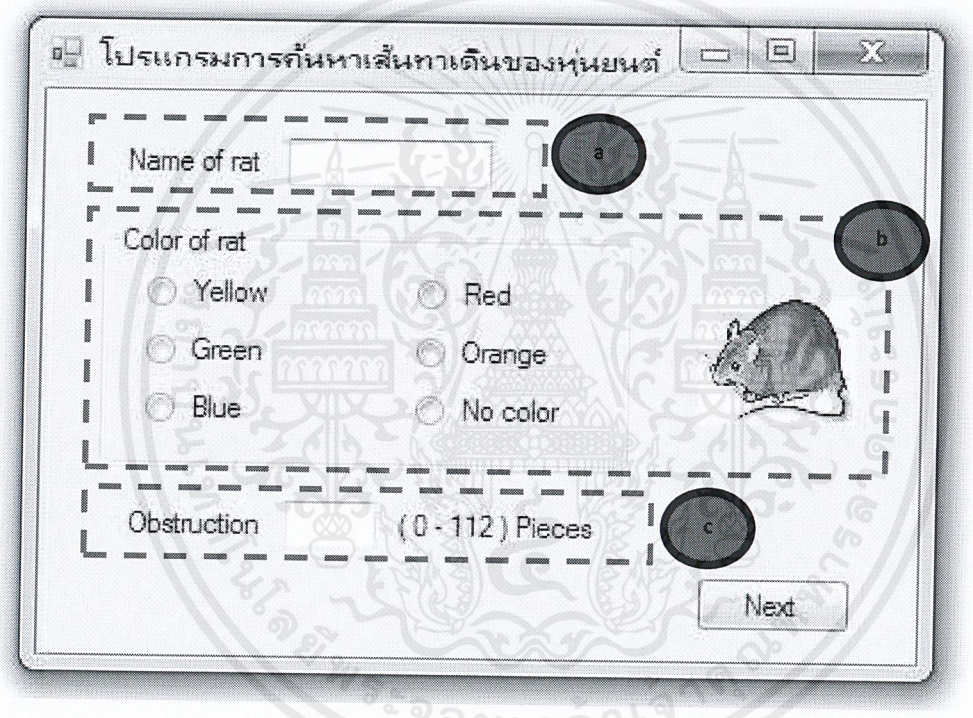
การออกแบบโปรแกรมนั้น จะต้องออกแบบให้สอดคล้องกับการจำลองการค้นหาเส้นทางเดินของหุ่นยนต์ โดยจะต้องคำนึงถึงการออกแบบขั้นตอนที่ใช้ประมวลผลข้อมูล การจัดเก็บข้อมูลที่ใช้ในการประมวลผล และการแสดงผลลัพธ์นั้นจะจัดเก็บไว้ภายใต้ตัวแปรใด จากนั้นจึงกำหนดขั้นตอนวิธีที่ใช้ในการแก้ปัญหา

การออกแบบโปรแกรมในส่วนของ Microsoft Visual Studio 2008 ประกอบด้วย 3 ส่วนหลักๆคือ

3.3.2.1 <u>การออกแบบฟอร์มนำเข้าข้อมูล</u>

การออกแบบฟอร์มนำเข้าข้อมูลที่ต้องทำการใส่ข้อมูล แบ่งได้เป็น 2 กรณีดังต่อไปนี้

1. กรณีของส่วนรับข้อมูลในฟอร์มที่ 1 ดังรูปที่ 3.10



รูปที่ 3.10 แสดงส่วนรับข้อมูลในฟอร์มที่ 1

จากรูปที่ 3.10 สามารถอธิบายรายละเอียดของแต่ละส่วนได้ดังต่อไปนี้

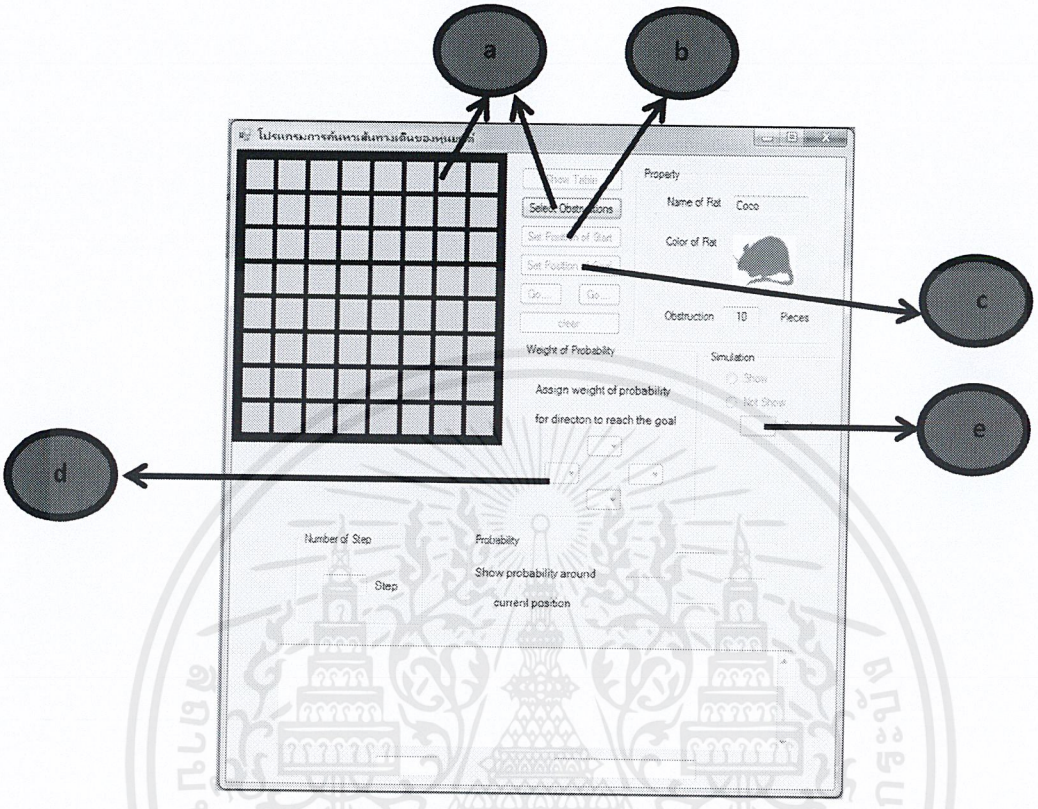
a) ชื่อของหนู (Name of rat) โดยผู้ใช้ต้องทำการกำหนดชื่อของหนู ซึ่งมีการกำหนดให้ใส่ตัวอักษรได้จำนวนไม่เกิน 10 ตัวอักษร

b) สีของหนู (Color of rat) โดยผู้ใช้สามารถเลือกสีตามความต้องการได้ และจะมีรูปภาพของหนูปรากฏอยู่ที่ด้านขวามือ ซึ่งจะมีสีตามที่ผู้ใช้เลือก

c) จำนวนสิ่งกีดขวาง (Obstruction) โดยผู้ใช้ต้องทำการกำหนดจำนวนสิ่งกีดขวาง และต้องอยู่ในช่วงที่กำหนดไว้ตั้งแต่ 0 – 112 ชิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กรณีของส่วนรับข้อมูลในฟอร์มที่ 2 ดังรูปที่ 3.11



รูปที่ 3.11 แสดงส่วนรับข้อมูลในฟอร์มที่ 2

จากรูปที่ 3.11 สามารถอธิบายรายละเอียดของแต่ละส่วนได้ดังต่อไปนี้

a) รับตำแหน่งสิ่งกีดขวาง ผู้ใช้สามารถเลือกตำแหน่งสิ่งกีดขวางตามจำนวนที่ผู้ใช้ได้เลือกไว้ในฟอร์มที่ 1 หลังจากที่ผู้ใช้ทำการกดเลือกที่ตำแหน่งใดๆ ตำแหน่งนั้นจะเปลี่ยนสีเป็นสีแดง เพื่อแสดงให้ผู้ใช้เห็นว่าได้ทำการเลือกตำแหน่งนั้นแล้ว

b) รับตำแหน่งเริ่มต้น ผู้ใช้สามารถเลือกตำแหน่งจุดเริ่มต้นได้

c) รับตำแหน่งเป้าหมาย ผู้ใช้สามารถเลือกตำแหน่งเป้าหมายได้

d) รับค่าน้ำหนักของความน่าจะเป็นในแต่ละทิศทาง เมื่อผู้ใช้ได้เลือกตำแหน่งเริ่มต้น และตำแหน่งเป้าหมายแล้ว ผู้ใช้จะรู้แนวโน้มทิศทางที่จะไปยังเป้าหมาย ทำให้ผู้ใช้สามารถเลือกได้ว่าจะให้ทิศทางใดมีค่าน้ำหนักของความน่าจะเป็นมากกว่า

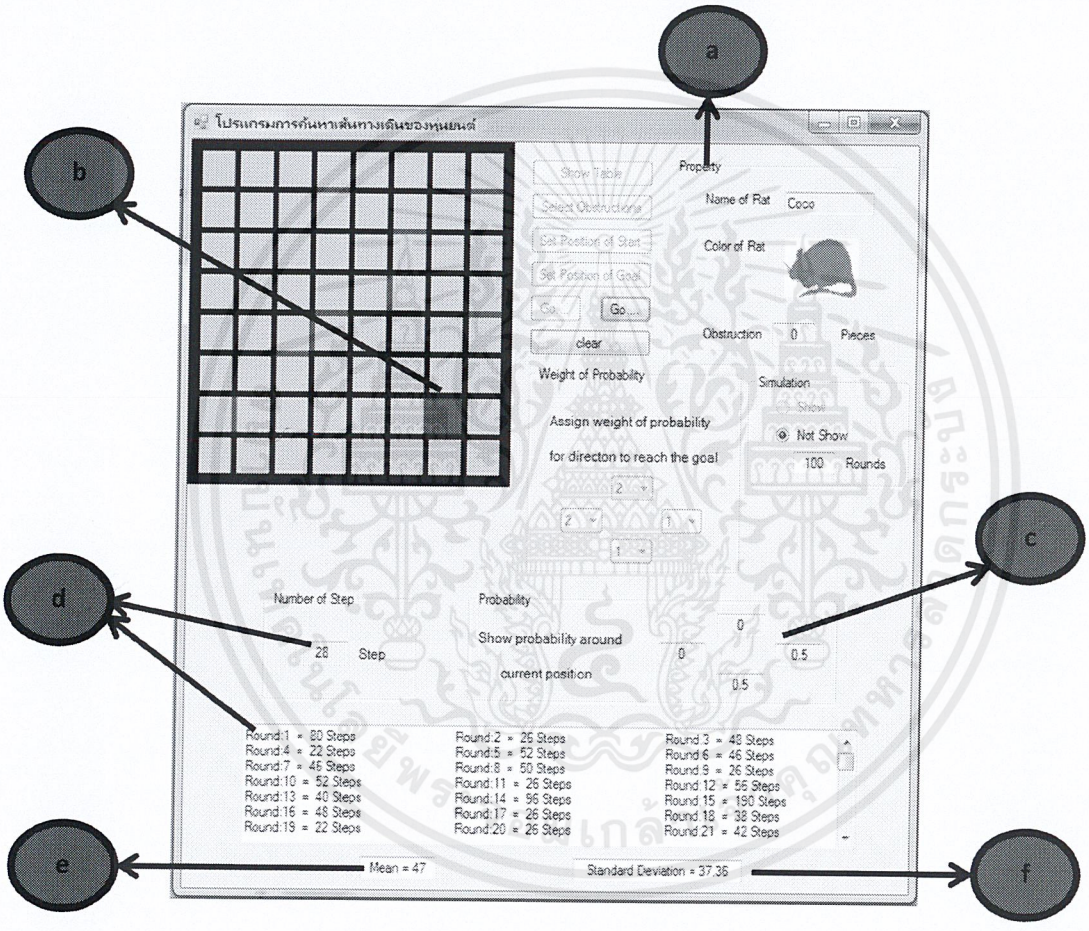
e) รับจำนวนรอบในการเดิน ในส่วนของ Simulation เมื่อผู้ใช้เลือกไม่แสดงการเคลื่อนที่ (Not Show) จะมีช่องว่างให้ผู้ใส่จำนวนรอบในการเดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 ส่วนประมวลผล (Data processing)

โดยการประมวลผลจะเริ่มจากการหาตำแหน่งปัจจุบัน จากนั้นจะทำการตรวจสอบทั้ง 4 ทิศทางรอบๆตำแหน่งนั้น เพื่อทำการกำหนดค่าความน่าจะเป็นของแต่ละทิศทาง ซึ่งในการกำหนดค่าความน่าจะเป็นในแต่ละทิศทาง จะคำนวณตามวิธีการคำนวณค่าความน่าจะเป็นของการเคลื่อนที่ และจะทำการเลือกทิศทางที่จะเคลื่อนที่ไป ด้วยวิธีการจำลองการเคลื่อนที่

3.3.2.3 ส่วนแสดงผล (Display)



รูปที่ 3.12 แสดงส่วนแสดงผล (Display)

จากรูปที่ 3.12 สามารถอธิบายรายละเอียดของแต่ละส่วนได้ดังต่อไปนี้

a) คุณสมบัติ (Property)

ในส่วนนี้จะเป็นการแสดงข้อมูล ที่ผู้ใช้ได้กำหนดไว้ในฟอร์มที่ 1 ซึ่งมีทั้งชื่อของหนู สีของหนู รูปภาพของหนู และจำนวนของสิ่งกีดขวาง โดยในส่วนของตัวเลขที่แสดงจำนวนสิ่งกีดขวางจะมีการลดลง เมื่อผู้ใช้ทำการกดเลือกตำแหน่งของสิ่งกีดขวาง และตัวเลขจะมีการเพิ่มขึ้น เมื่อผู้ใช้กดตำแหน่งเดิมที่ได้กดไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b) แสดงการเคลื่อนที่ (Movement)

ในส่วนนี้จะเป็นการแสดงการเคลื่อนที่จากตำแหน่งเริ่มต้นไปยังตำแหน่งเป้าหมาย โดยในระหว่างการเคลื่อนที่ตำแหน่งที่เป็นตำแหน่งปัจจุบันจะถูกเปลี่ยนให้เป็นสีน้ำตาล และเมื่อเคลื่อนที่ไปยังตำแหน่งต่อไป ตำแหน่งก่อนหน้าจะถูกเปลี่ยนให้เป็นสีฟ้าดังเดิม

c) แสดงค่าความน่าจะเป็น (Probability)

ในส่วนนี้จะเป็นการแสดงค่าความน่าจะเป็นของแต่ละทิศทางรอบๆตำแหน่งปัจจุบันที่ได้จากการประมวลผล

d) แสดงจำนวนก้าว (Number of step)

ในส่วนนี้จะเป็นการแสดงจำนวนก้าว ที่เดินจากตำแหน่งเริ่มต้น ไปยังตำแหน่งเป้าหมาย ซึ่งจะแบ่งออกเป็น 2 ส่วน โดยส่วนที่ 1 จะเป็นการแสดงจำนวนก้าว ตั้งแต่ก้าวแรกจนถึงก้าวสุดท้ายที่ทำการเดิน ส่วนที่ 2 จะเป็นการแสดงผลลัพธ์ หรือจำนวนก้าวที่เกิดขึ้นในแต่ละรอบของการเดิน

e) แสดงค่าเฉลี่ยตัวอย่าง

ในส่วนนี้จะเป็นการแสดงค่าเฉลี่ยของจำนวนก้าว ซึ่งสามารถคำนวณได้จาก วิธีการคำนวณค่าเฉลี่ยตัวอย่าง (Sample Average, $\bar{x}$)

f) แสดงค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง

ในส่วนนี้จะเป็นการแสดงค่าส่วนเบี่ยงเบนมาตรฐานของจำนวนก้าว ซึ่งสามารถคำนวณได้จาก วิธีการคำนวณค่าส่วนเบี่ยงเบนมาตรฐาน

3.3.3 การเขียนโปรแกรม

การเขียนโปรแกรมเป็นการนำผลลัพธ์ของการออกแบบโปรแกรมมาเปลี่ยนเป็นภาษาคอมพิวเตอร์ ในปริญญานิพนธ์ฉบับนี้ใช้ภาษา Visual C# ตัวอย่างการเขียนโปรแกรมค้นหาเส้นทางเดินของหุ่นยนต์แสดงในภาคผนวก ข

3.3.4 การทดสอบและแก้ไขโปรแกรม

การทดสอบโปรแกรมเป็นการนำโปรแกรมที่เขียนโค้ดแล้วเข้าคอมพิวเตอร์ เพื่อตรวจสอบรูปแบบกฎเกณฑ์ของภาษาและผลการทำงานของผลโปรแกรมนั้น ซึ่งเป็นการจำลองการเคลื่อนที่ของหุ่นยนต์บนคอมพิวเตอร์ในสนามจำลองขนาด 8x8 ถ้ามีการตรวจพบว่ายังไม่ถูกต้องให้ดำเนินการแก้ไขให้ถูกต้องต่อไป

3.3.5 การนำโปรแกรมไปใช้งาน

โปรแกรมสามารถนำไปใช้งานในการค้นหาเส้นทางเดินของหุ่นยนต์ โดยที่ไม่ทราบตำแหน่งเป้าหมายล่วงหน้า และสามารถหลบหลีกสิ่งกีดขวางที่ถูกวางไว้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการดำเนินงาน

หลังจากที่ทำการสร้างโปรแกรมในการค้นหาเส้นทางเดินของหุ่นยนต์เสร็จแล้ว ได้มีการทดลอง เพื่อหาจำนวนรอบในการเดิน เพื่อใช้เป็นขนาดตัวอย่างในการศึกษาว่ามีตัวแปรใดบ้าง ที่มีผลต่อจำนวนการเคลื่อนที่ของหุ่นยนต์ ซึ่งตัวแปรที่จะนำมาศึกษา คือ จำนวนสิ่งกีดขวาง และน้ำหนักที่ให้กับค่าความน่าจะเป็น

โดยในการทดลองเพื่อศึกษาว่าจำนวนสิ่งกีดขวางมีผลต่อจำนวนก้าวหรือไม่ จะใช้สิ่งกีดขวางจำนวน 10 ชิ้น และ 15 ชิ้น แล้วใช้วิธีการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง เพื่อหาผลต่างระหว่างจำนวนก้าวโดยเฉลี่ย ของการใช้สิ่งกีดขวางจำนวน 10 ชิ้น และ 15 ชิ้น

ส่วนในการทดลองเพื่อศึกษาว่าน้ำหนักที่ให้กับค่าความน่าจะเป็น มีผลต่อจำนวนก้าวหรือไม่ จะใช้การให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด และการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า แล้วใช้วิธีการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง เพื่อหาผลต่างระหว่างจำนวนก้าวโดยเฉลี่ย ของการให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด และการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า

4.1 การทดลองเพื่อหาจำนวนรอบในการเดิน

ก่อนที่จะทำการทดลองเพื่อศึกษาว่ามีตัวแปรใดบ้าง ที่มีผลต่อจำนวนการเคลื่อนที่ ได้ทำการทดลองเพื่อหาจำนวนรอบในการเดิน หรือขนาดตัวอย่าง (n) ที่จะใช้ในการทดลอง เนื่องจากการเลือกกลุ่มตัวอย่างในการทดลอง นับว่าเป็นขั้นตอนที่มีความสำคัญอีกขั้นตอนหนึ่ง เพราะเป็นการเลือกตัวแทนมาศึกษา ผลจากการศึกษาที่ได้ จะต้องใช้สรุปอ้างอิงถึงประชากร

โดยการทดลองจะเริ่มจากการใช้ 125 รอบ หรือขนาดตัวอย่างเท่ากับ 125 จากนั้นทำการเพิ่มจำนวนรอบขึ้นครั้งละ 2 เท่า โดยเพิ่มขึ้นไปเรื่อยๆ จนกว่าเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) ไม่เกิน 5% ส่วนจำนวนสิ่งกีดขวาง จำนวนรอบในการเดิน น้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน ตำแหน่งสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมายเหมือนเดิม

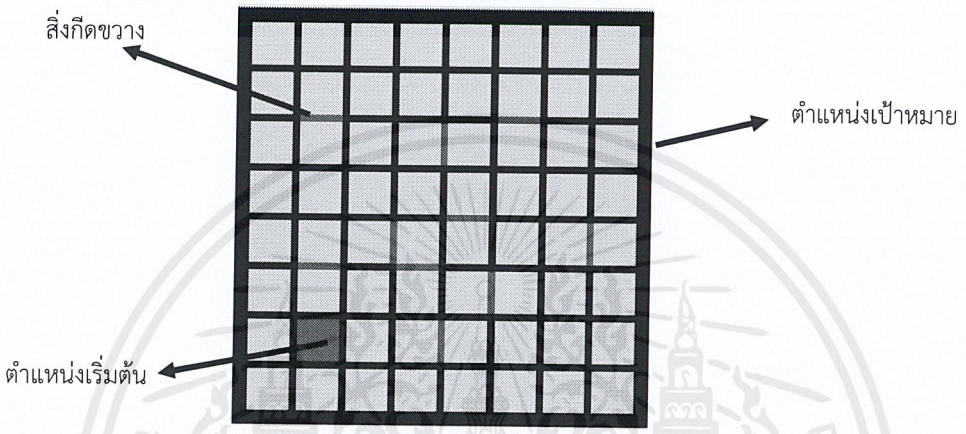
- จำนวน 125 รอบ

ในการทดลองกำหนดให้ใช้จำนวนสิ่งกีดขวาง จำนวนรอบในการเดิน และน้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน ดังตารางที่ 4.1 และตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย แสดงดังรูปที่ 4.1

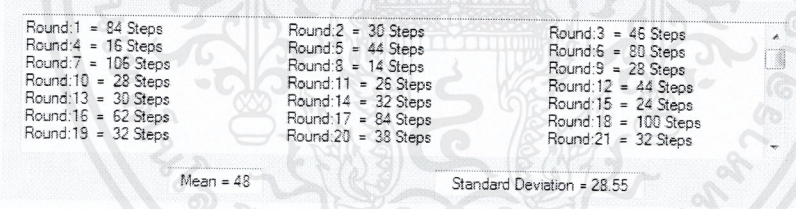
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยใช้ 125 รอบ

จำนวนสิ่งกีดขวาง (ชิ้น)	จำนวนรอบในการเดิน (รอบ)	น้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน			
		ด้านบน	ด้านล่าง	ด้านซ้าย	ด้านขวา
10	125	2	1	1	2



รูปที่ 4.1 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยใช้ 125 รอบ



รูปที่ 4.2 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 125 รอบ

จากการประมวลผล ดังรูปที่ 4.2 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดินเท่ากับ 48 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 28.55 จากนั้นนำทั้ง 2 ค่ามาทำการประมาณค่าเฉลี่ยประชากรเดียวแบบช่วง ดังนี้

ช่วงความเชื่อมั่น 95% ของ μ เมื่อไม่ทราบค่า σ^2 และ $n \geq 30$ คือ

$$\bar{x} - z_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}} < \mu < \bar{x} + z_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}}$$

$$48 - 1.96\frac{28.55}{\sqrt{125}} < \mu < 48 + 1.96\frac{28.55}{\sqrt{125}}$$

$$42.995 < \mu < 53.005$$

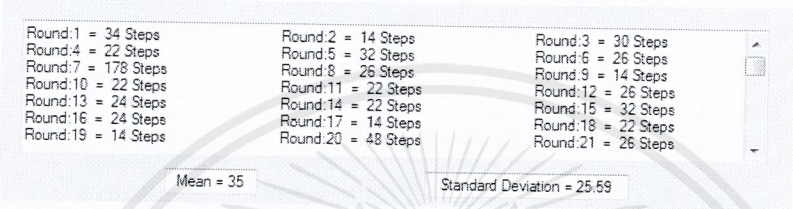
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ

$$48 \pm 5.005$$

ดังนั้น ช่วงความเชื่อมั่น 95% ของจำนวนก้าวโดยเฉลี่ยที่แท้จริงอยู่ในช่วง 42.995 ถึง 53.005 ก้าว และเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) $= \pm 10.427\%$

- จำนวน 250 รอบ



รูปที่ 4.3 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 250 รอบ

จากการประมวลผล ดังรูปที่ 4.3 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดินเท่ากับ 35 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 25.59 จากนั้นนำทั้ง 2 ค่ามาทำการประมาณค่าเฉลี่ยประชากรเดียวแบบช่วง ดังนี้

ช่วงความเชื่อมั่น 95% ของ μ เมื่อไม่ทราบค่า σ^2 และ $n \geq 30$ คือ

$$\bar{x} - z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}} < \quad \mu \quad < \bar{x} + z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}$$

$$35 - 1.96 \frac{25.59}{\sqrt{250}} < \quad \mu \quad < 35 + 1.96 \frac{25.59}{\sqrt{250}}$$

$$31.828 < \quad \mu \quad < 38.172$$

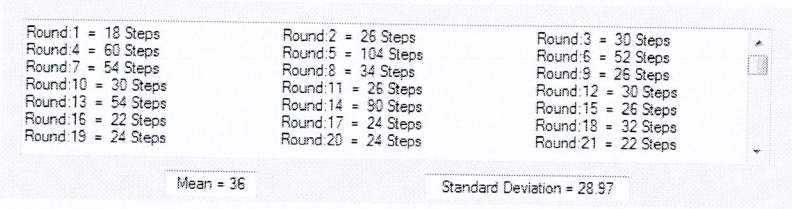
หรือ

$$35 \pm 3.172$$

ดังนั้น ช่วงความเชื่อมั่น 95% ของจำนวนก้าวโดยเฉลี่ยที่แท้จริงอยู่ในช่วง 31.828 ถึง 38.172 ก้าว และเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) $= \pm 9.063\%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จำนวน 500 รอบ



รูปที่ 4.4 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 500 รอบ

จากการประมวลผล ดังรูปที่ 4.4 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดินเท่ากับ 36 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 28.97 จากนั้นนำทั้ง 2 ค่ามาทำการประมาณค่าเฉลี่ยประชากรเดียวแบบช่วง ดังนี้

ช่วงความเชื่อมั่น 95% ของ μ เมื่อไม่ทราบค่า σ^2 และ $n \geq 30$ คือ

$$\bar{x} - z_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}} < \mu < \bar{x} + z_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}}$$

$$36 - 1.96\frac{28.97}{\sqrt{500}} < \mu < 36 + 1.96\frac{28.97}{\sqrt{500}}$$

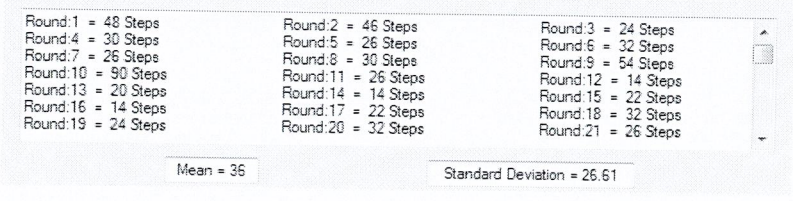
$$33.461 < \mu < 38.539$$

หรือ

$$36 \pm 2.539$$

ดังนั้น ช่วงความเชื่อมั่น 95% ของจำนวนก้าวโดยเฉลี่ยที่แท้จริงอยู่ในช่วง 33.461 ถึง 38.539 ก้าว และเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) $= \pm 7.053\%$

- จำนวน 1000 รอบ



รูปที่ 4.5 แสดงผลลัพธ์ที่ได้จากการประมวลผล โดยใช้ 1000 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการประมวลผล ดังรูปที่ 4.5 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดิน เท่ากับ 36 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 26.61 จากนั้นนำทั้ง 2 ค่ามาทำการประมาณค่าเฉลี่ยประชากรเดียวแบบช่วง ดังนี้

ช่วงความเชื่อมั่น 95% ของ μ เมื่อไม่ทราบค่า σ^2 และ $n \geq 30$ คือ

$$\bar{x} - z_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}} < \quad \mu \quad < \bar{x} + z_{\frac{\alpha}{2}}\frac{s}{\sqrt{n}}$$

$$36 - 1.96\frac{26.62}{\sqrt{1000}} < \quad \mu \quad < 36 + 1.96\frac{26.62}{\sqrt{1000}}$$

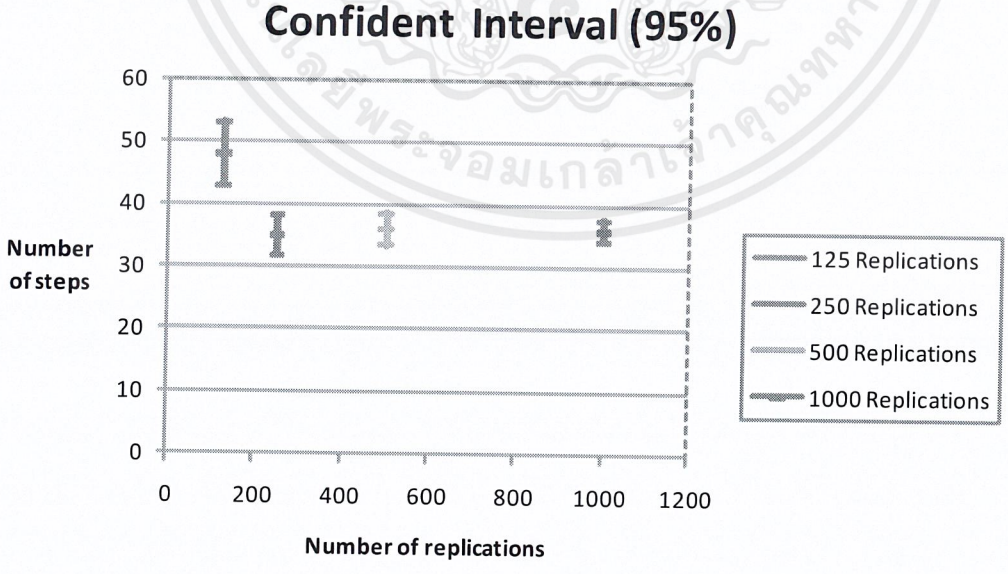
$$34.351 < \quad \mu \quad < 37.649$$

หรือ

$$36 \pm 1.649$$

ดังนั้น ช่วงความเชื่อมั่น 95% ของจำนวนก้าวโดยเฉลี่ยที่แท้จริงอยู่ในช่วง 34.351 ถึง 37.649 ก้าว และเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) $= \pm 4.581\%$

สรุปผลจากการทดลอง จึงเลือกใช้จำนวนรอบในการเดินเป็นจำนวน 1000 รอบ ในการทดลองเปลี่ยนจำนวนสิ่งกีดขวาง และทดลองเปลี่ยนน้ำหนักที่ให้กับค่าความน่าจะเป็น เนื่องจากเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) $= \pm 4.581\%$ ซึ่งไม่เกิน 5% ตามที่กำหนดไว้ และจากรูปที่ 4.6 จะเห็นได้ว่าช่วงความเชื่อมั่น 95% ของ μ มีช่วงที่แคบลงเรื่อยๆ



รูปที่ 4.6 แสดงช่วงความเชื่อมั่น 95% ของ μ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองเปลี่ยนจำนวนสิ่งกีดขวาง

ในการทดลองเพื่อศึกษาว่าจำนวนสิ่งกีดขวางมีผลต่อจำนวนก้าวหรือไม่ ได้ใช้วิธีการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง

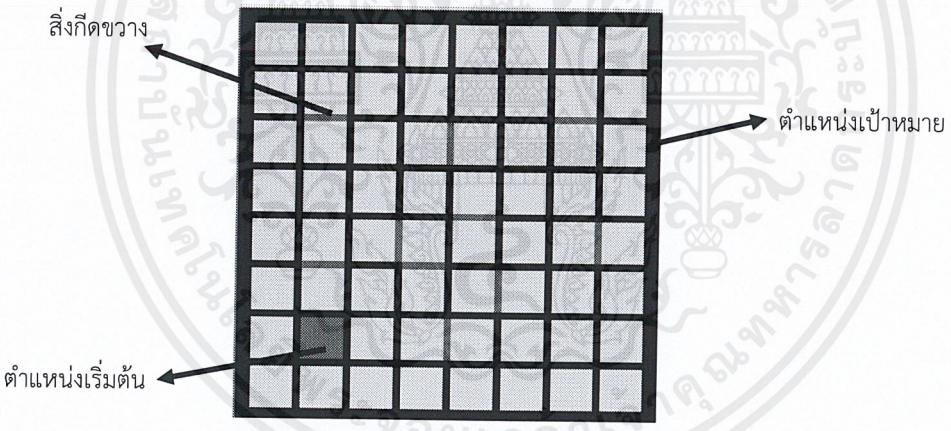
ดังนั้นจึงเลือกที่จะทดลองโดยใช้สิ่งกีดขวางจำนวน 10 ชิ้น และ 15 ชิ้น ส่วนตำแหน่งของสิ่งกีดขวางนั้น เมื่อมีการเพิ่มจำนวนสิ่งกีดขวาง จะยึดตำแหน่งเดิม แล้วทำการเพิ่มตำแหน่งตามจำนวนสิ่งกีดขวางที่เพิ่มขึ้น

- สิ่งกีดขวางจำนวน 10 ชิ้น

ในการทดลองกำหนดให้ใช้จำนวนสิ่งกีดขวาง จำนวนรอบในการเดิน และน้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน ดังตารางที่ 4.2 และตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย แสดงดังรูปที่ 4.7

ตารางที่ 4.2 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยใช้สิ่งกีดขวาง 10 ชิ้น

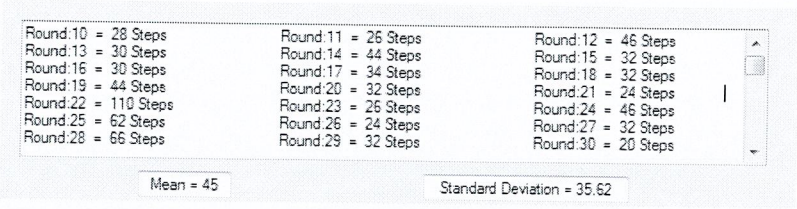
จำนวนสิ่งกีดขวาง (ชิ้น)	จำนวนรอบในการเดิน (รอบ)	น้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน			
		ด้านบน	ด้านล่าง	ด้านซ้าย	ด้านขวา
10	1000	2	1	1	2



รูปที่ 4.7 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยใช้สิ่งกีดขวาง 10 ชิ้น

เมื่อกดปุ่ม Go โปรแกรมจะทำการประมวลผล จากการประมวลผล ดังรูปที่ 4.8 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดิน เท่ากับ45 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 35.62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



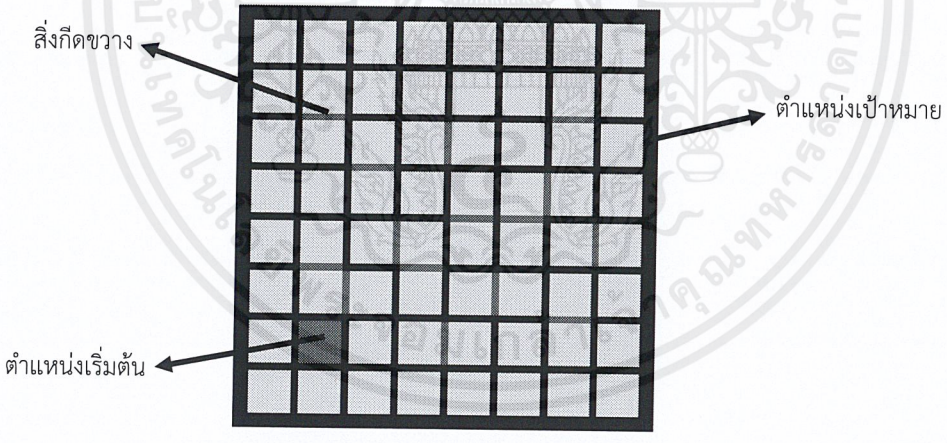
รูปที่ 4.8 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง โดยใช้สิ่งกีดขวาง 10 ชิ้น

- สิ่งกีดขวางจำนวน 15 ชิ้น

ในการทดลองกำหนดให้ใช้จำนวนสิ่งกีดขวาง จำนวนรอบในการเดิน และน้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน ดังตารางที่ 4.3 และตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย แสดงดังรูปที่ 4.9

ตารางที่ 4.3 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยใช้สิ่งกีดขวาง 15 ชิ้น

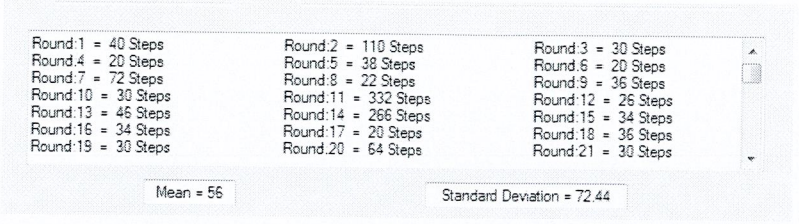
จำนวนสิ่งกีดขวาง (ชิ้น)	จำนวนรอบในการเดิน (รอบ)	น้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน			
		ด้านบน	ด้านล่าง	ด้านซ้าย	ด้านขวา
15	1000	2	1	1	2



รูปที่ 4.9 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยใช้สิ่งกีดขวาง 15 ชิ้น

เมื่อกดปุ่ม Go โปรแกรมจะทำการประมวลผล จากการประมวลผล ดังรูปที่ 4.10 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดินเท่ากับ 56 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 72.44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง โดยใช้สิ่งกีดขวาง 15 ชิ้น

หลังจากที่ทำการทดลองโดยใช้สิ่งกีดขวางจำนวน 10 ชิ้น และ 15 ชิ้น จะได้ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่างของทั้ง 2 การทดลอง ซึ่งสามารถนำมาใช้ในการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง ดังนี้

ให้ μ_1 แทน จำนวนก้าวโดยเฉลี่ยของการใช้สิ่งกีดขวางจำนวน 10 ชิ้น

ให้ μ_2 แทน จำนวนก้าวโดยเฉลี่ยของการใช้สิ่งกีดขวางจำนวน 15 ชิ้น

ช่วงความเชื่อมั่น 95% ของ $\mu_1 - \mu_2$ เมื่อไม่ทราบค่า σ_1^2 , σ_2^2และ $n_1 \geq 30$, $n_2 \geq 30$ คือ

$$(\bar{x}_1 - \bar{x}_2) - z_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} < \quad \mu_1 - \mu_2 \quad < (\bar{x}_1 - \bar{x}_2) + z_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

$$(45 - 56) - 1.96\sqrt{\frac{35.62^2}{1000} + \frac{72.44^2}{1000}} < \quad \mu_1 - \mu_2 \quad < (45 - 56) + 1.96\sqrt{\frac{35.62^2}{1000} + \frac{72.44^2}{1000}}$$

$$-16.003 < \quad \mu_1 - \mu_2 \quad < -5.997$$

ช่วงความเชื่อมั่น 95% สำหรับผลต่างระหว่าง จำนวนก้าวโดยเฉลี่ยของการใช้สิ่งกีดขวางจำนวน 10 ชิ้น และ 15 ชิ้น อยู่ในช่วง -16.003 ถึง -5.997 ก้าว

จะเห็นว่า จำนวนก้าวโดยเฉลี่ยของการใช้สิ่งกีดขวางจำนวน 15 ชิ้น มากกว่าจำนวนก้าวโดยเฉลี่ยของการใช้สิ่งกีดขวางจำนวน 10 ชิ้น ระหว่าง 5.997 ถึง 16.003 ก้าว

ดังนั้น การเปลี่ยนจำนวนสิ่งกีดขวางมีผลต่อจำนวนก้าว โดยในการทดลองนี้เป็นการเพิ่มจำนวนของสิ่งกีดขวาง ซึ่งมีผลทำให้จำนวนก้าวเพิ่มขึ้น

4.3 การทดลองเปลี่ยนน้ำหนักที่ให้กับค่าความน่าจะเป็น

ในการทดลองเพื่อศึกษาว่าน้ำหนักที่ให้กับค่าความน่าจะเป็น มีผลต่อจำนวนก้าวหรือไม่ ได้ใช้วิธีการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

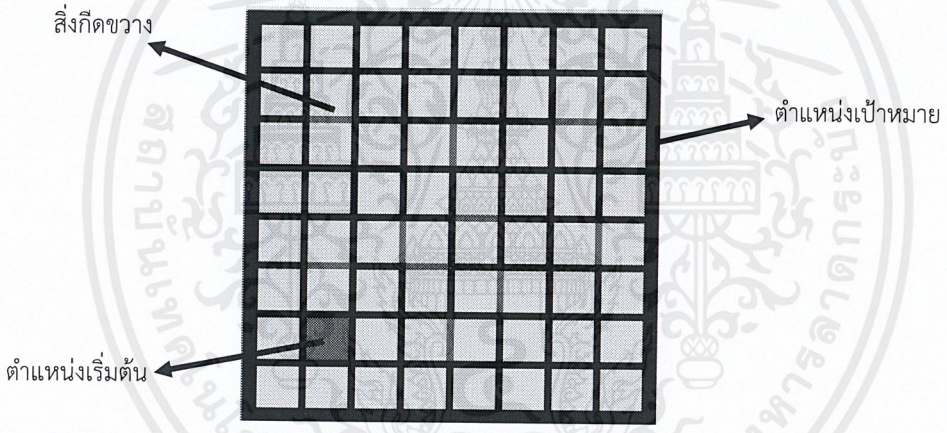
ดังนั้นจะทำการเปรียบเทียบระหว่างการให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด กับการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า โดยปัจจัยที่กำหนดให้คงที่มีจำนวนสิ่งกีดขวาง ตำแหน่งสิ่งกีดขวาง ตำแหน่งเริ่มต้น ตำแหน่งเป้าหมาย จำนวนรอบในการเดิน

- กรณีการให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด

ในการทดลองกำหนดให้ใช้จำนวนสิ่งกีดขวาง จำนวนรอบในการเดิน และน้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน ดังตารางที่ 4.4 และตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย แสดงดังรูปที่ 4.11

ตารางที่ 4.4 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด

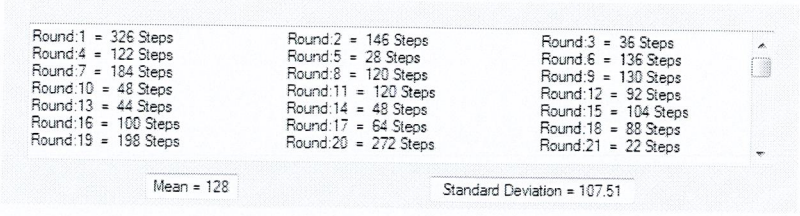
จำนวนสิ่งกีดขวาง (ชิ้น)	จำนวนรอบในการเดิน (รอบ)	น้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน			
		ด้านบน	ด้านล่าง	ด้านซ้าย	ด้านขวา
10	1000	1	1	1	1



รูปที่ 4.11 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด

เมื่อกดปุ่ม Go โปรแกรมจะทำการประมวลผล จากการประมวลผล ดังรูปที่ 4.12 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดินเท่ากับ 128 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 107.51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



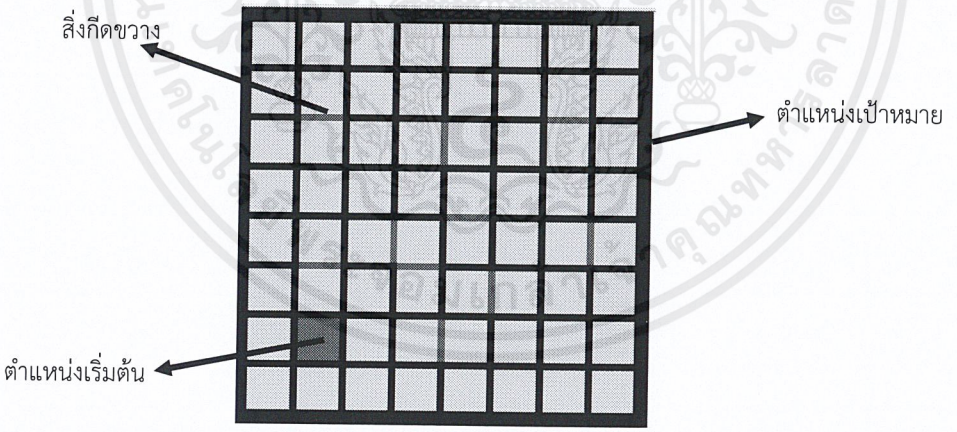
รูปที่ 4.12 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง โดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด

- กรณีการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า

ในการทดลองกำหนดให้ใช้จำนวนสิ่งกีดขวาง จำนวนรอบในการเดิน และน้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน ดังตารางที่ 4.5 และตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย แสดงดังรูปที่ 4.13

ตารางที่ 4.5 แสดงการกำหนดค่าต่างๆที่ใช้ในการทดลอง โดยให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า

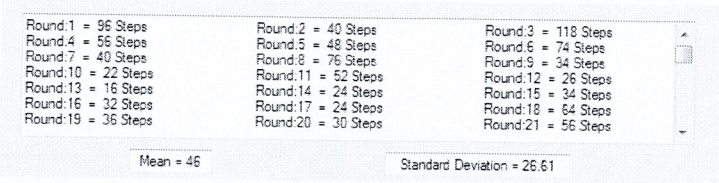
จำนวนสิ่งกีดขวาง (ชิ้น)	จำนวนรอบในการเดิน (รอบ)	น้ำหนักที่ให้กับค่าความน่าจะเป็นในแต่ละด้าน			
		ด้านบน	ด้านล่าง	ด้านซ้าย	ด้านขวา
10	1000	2	1	1	2



รูปที่ 4.13 แสดงตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น และตำแหน่งเป้าหมาย โดยให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า

เมื่อกดปุ่ม Go โปรแกรมจะทำการประมวลผล จากการประมวลผล ดังรูปที่ 4.14 ได้ค่าเฉลี่ยตัวอย่างของจำนวนก้าวในการเดินเท่ากับ 46 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 26.61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงจำนวนก้าวในแต่ละรอบ ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่าง โดยให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า

หลังจากที่ทำการทดลองโดยมีการให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด และการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า จะได้ค่าเฉลี่ยตัวอย่าง และส่วนเบนมาตรฐานตัวอย่างของทั้ง 2 การทดลอง ซึ่งสามารถนำมาใช้ในการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง ดังนี้

ให้ μ_1 แทน จำนวนก้าวโดยเฉลี่ยโดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด

ให้ μ_2 แทน จำนวนก้าวโดยเฉลี่ยโดยให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า

ช่วงความเชื่อมั่น 95% ของ $\mu_1 - \mu_2$ เมื่อไม่ทราบค่า σ_1^2 , σ_2^2 และ $n_1 \geq 30$, $n_2 \geq 30$ คือ

$$(\bar{x}_1 - \bar{x}_2) - z_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} < \quad \mu_1 - \mu_2 \quad < (\bar{x}_1 - \bar{x}_2) + z_{\frac{\alpha}{2}}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

$$(128 - 46) - 1.96\sqrt{\frac{107.51^2}{1000} + \frac{26.61^2}{1000}} < \quad \mu_1 - \mu_2 \quad < (128 - 46) + 1.96\sqrt{\frac{107.51^2}{1000} + \frac{26.61^2}{1000}}$$

$$75.135 < \quad \mu_1 - \mu_2 \quad < 88.865$$

ช่วงความเชื่อมั่น 95% สำหรับผลต่างระหว่าง จำนวนก้าวโดยเฉลี่ยโดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด และให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า อยู่ในช่วง 75.135 ถึง 88.865 ก้าว

จะเห็นว่า จำนวนก้าวโดยเฉลี่ยโดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด มากกว่าจำนวนก้าวโดยเฉลี่ยของการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า ดังนั้น การเปลี่ยนน้ำหนักที่ให้กับค่าความน่าจะเป็นมีผลต่อจำนวนก้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัย

5.1 สรุปผลการดำเนินงาน

ปริญญานิพนธ์เรื่องการออกแบบและพัฒนาโปรแกรมในการค้นหาเส้นทางเดินของหุ่นยนต์ โดยมีวัตถุประสงค์เพื่อประยุกต์ใช้กระบวนการตัดสินใจมาร์คอฟ ซึ่งเป็นกระบวนการสุ่มที่มีคุณสมบัติคือ สถานะในอนาคตของกระบวนการขึ้นอยู่กับสถานะปัจจุบันเท่านั้น

จากการศึกษาความเป็นไปได้ของการนำทฤษฎีของมาร์คอฟ มาใช้ค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้ตารางขนาด 3x3 มาทำการทดลอง และทำการคำนวณโดยใช้เมตริกซ์เดิมคูณตัวเอง เป็นจำนวนครั้งที่ทำให้เข้าสู่สถานะคงที่ สามารถแบ่งได้ออกเป็น 2 กรณี คือ กรณีที่ 1 เป็นการกำหนดตำแหน่งสิ่งกีดขวาง ให้มีช่องทางที่สามารถไปยังเป้าหมายได้ ซึ่งจากผลการคำนวณที่ n เท่ากับ150 มีความน่าจะเป็นของการเปลี่ยนสถานะเท่ากับ 0.0741 > 0 ซึ่งแสดงให้เห็นว่ามีความน่าจะเป็นที่จะไปยังเป้าหมายได้ กรณีที่ 2 เป็นการกำหนดตำแหน่งสิ่งกีดขวาง ไม่ให้มีช่องทางที่สามารถไปยังเป้าหมายได้ซึ่งจากผลการคำนวณที่ n เท่ากับ150 มีความน่าจะเป็นของการเปลี่ยนสถานะเท่ากับ 0 ซึ่งแสดงให้เห็นว่าไม่มีความน่าจะเป็นที่จะไปยังเป้าหมายได้

จากการศึกษาปัจจัยที่มีผลต่อการปรับเปลี่ยนสถานะโดยใช้ตารางขนาด 3x3 มาทำการทดลอง โดยมีการให้ทางที่จะไปยังเป้าหมาย มีค่าความน่าจะเป็นมากกว่าทางที่เหลือ 1.5 เท่า ซึ่งจากผลการคำนวณที่ n เท่ากับ150 มีความน่าจะเป็นของการเปลี่ยนสถานะเท่ากับ 0.4826 ส่วนการให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด ผลการคำนวณที่ n เท่ากับ 150 มีความน่าจะเป็นของการเปลี่ยนสถานะเท่ากับ 0.0741 จึงสามารถสรุปได้ว่า การให้น้ำหนักมากหรือน้อยกับค่าความน่าจะเป็นของแต่ละทิศทาง มีผลต่อค่าความน่าจะเป็นของการเปลี่ยนสถานะ

หลังจากที่ทำการสร้างโปรแกรมในการค้นหาเส้นทางเดินของหุ่นยนต์เสร็จแล้ว ได้ทำการทดลองเพื่อหาจำนวนรอบในการเดิน หรือขนาดตัวอย่าง (n) ที่จะใช้ในการทดลอง โดยการทดลองจะเริ่มจากการใช้ 125 รอบ หรือขนาดตัวอย่างเท่ากับ 125 จากนั้นทำการเพิ่มจำนวนรอบขึ้นครั้งละ 2 เท่า โดยเพิ่มขึ้นไปเรื่อยๆ จนกว่าเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) ไม่เกิน 5% สามารถสรุปผลได้ดังตารางที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.1 แสดงการสรุปค่าของเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate)

จำนวนรอบ	ขอบเขตของความผิดพลาด (Margin of error)
125	$\pm$10.427%
250	$\pm$*9.063%*
500	$\pm$*7.053%*
1000	$\pm$*4.581%*

จากผลการทดลองดังตารางที่ 5.1 ได้เลือกใช้จำนวนรอบในการเดินเป็นจำนวน 1000 รอบ เนื่องจากเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ (Percent of Error in Estimate) = $\pm$*4.581%* ซึ่งไม่เกิน 5% ตามที่กำหนดไว้

หลังจากที่ได้ทำการทดลองเพื่อหาจำนวนรอบในการเดิน หรือขนาดตัวอย่าง (n) ที่ใช้ในการทดลองแล้ว ได้มีการนำโปรแกรมที่พัฒนาขึ้นมาทำการทดลองต่อเพื่อศึกษาว่ามีตัวแปรใดบ้าง ที่มีผลต่อจำนวนการเคลื่อนที่

5.2 สรุปผลการนำโปรแกรมที่พัฒนาขึ้นมาใช้ทดลองเพื่อศึกษาตัวแปร ที่มีผลต่อการเคลื่อนที่

สามารถแบ่งการทดลองเพื่อศึกษาตัวแปร ที่มีผลต่อการเคลื่อนที่ได้ ดังนี้คือ

5.2.1 การทดลองเปลี่ยนจำนวนสิ่งกีดขวาง

ได้ใช้วิธีการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง โดยเลือกใช้สิ่งกีดขวางจำนวน 10 ชิ้น และ 15 ชิ้น ได้ผลการทดลองดังตารางที่ 5.2

ตารางที่ 5.2 แสดงค่าเฉลี่ยตัวอย่างและส่วนเบี่ยงเบนมาตรฐานตัวอย่าง ในการทดลองเปลี่ยนจำนวนสิ่งกีดขวาง

จำนวนสิ่งกีดขวาง	ค่าเฉลี่ยตัวอย่าง	ส่วนเบี่ยงเบนมาตรฐานตัวอย่าง
10	45	35.62
15	56	72.44

นำผลการทดลองดังตารางที่ 5.2 มาใช้ในการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง จะได้ว่าช่วงความเชื่อมั่น 95% สำหรับผลต่างระหว่าง จำนวนก้าวโดยเฉลี่ยของการใช้สิ่งกีดขวางจำนวน 10 ชิ้น และ 15 ชิ้น อยู่ในช่วง -16.003 ถึง -5.997 ก้าว

สามารถสรุปได้ว่า การเปลี่ยนจำนวนสิ่งกีดขวางมีผลต่อจำนวนก้าว โดยในการทดลองนี้เป็นการเพิ่มจำนวนของสิ่งกีดขวาง ซึ่งมีผลทำให้จำนวนก้าวเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การทดลองเปลี่ยนน้ำหนักที่ให้กับความน่าจะเป็น

ได้ใช้วิธีการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง โดยทำการเปรียบเทียบระหว่างการให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด กับการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า ได้ผลการทดลองดังตารางที่ 5.3

ตารางที่ 5.3 แสดงค่าเฉลี่ยตัวอย่างและส่วนเบี่ยงเบนมาตรฐานตัวอย่าง ในการทดลองเปลี่ยนน้ำหนักที่ให้กับความน่าจะเป็น

รูปแบบการให้น้ำหนัก	ค่าเฉลี่ยตัวอย่าง	ส่วนเบี่ยงเบนมาตรฐานตัวอย่าง
ให้น้ำหนัก 4 ทิศทางเท่ากันหมด	128	107.51
ให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า	46	26.61

นำผลการทดลองดังตารางที่ 5.3 มาใช้ในการประมาณค่าผลต่างระหว่างค่าเฉลี่ยประชากรของสองประชากรแบบช่วง จะได้ว่าช่วงความเชื่อมั่น 95% สำหรับผลต่างระหว่าง จำนวนก้าวโดยเฉลี่ยโดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด และให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า อยู่ในช่วง 75.135 ถึง 88.865 ก้าว

สามารถสรุปได้ว่า จำนวนก้าวโดยเฉลี่ยโดยให้น้ำหนักทั้ง 4 ทิศทางเท่ากันหมด มากกว่าจำนวนก้าวโดยเฉลี่ยของการให้น้ำหนักใน 2 ทิศทางที่จะไปยังเป้าหมายมีค่ามากกว่า ดังนั้น การเปลี่ยนน้ำหนักที่ให้กับค่าความน่าจะเป็นมีผลต่อจำนวนก้าว

5.3 แนวทางในการนำไปพัฒนา

เนื่องจากวิธีในการเขียนโปรแกรมในการหาเส้นทางเดินของหุ่นยนต์นั้น เป็นวิธีที่อาศัยหลักของความน่าจะเป็น จึงทำให้มีเส้นทางที่จะเดินไปยังเป้าหมายนั้นมีมากกว่า 1 เส้นทาง และอาจจะไม่ได้เป็นเส้นทางเดิมที่เคยไปยังเป้าหมาย จึงเป็นข้อดี คือ ถึงแม้ว่าจะมีจำนวนสิ่งกีดขวางและตำแหน่งมากน้อยเพียงใด หุ่นยนต์ก็จะมีโอกาสที่จะเดินไปถึงเป้าหมายได้ แต่ในขณะเดียวกันก็มีข้อเสีย คือ ในการเดินไปยังเป้าหมายบางครั้ง อาจมีจำนวนก้าวในการเดินไม่แน่นอน ซึ่งส่งผลต่อส่วนเบี่ยงเบนมาตรฐานของการทดลองที่มีค่ามาก อันเนื่องมาจากเกิดการแปรผันของจำนวนก้าวมาก

1. ดังนั้น ในการนำไปใช้งานจริงสำหรับการกู้ภัย จึงจำเป็นที่จะต้องมีการพัฒนาเพิ่มเติมในส่วนต่างๆ คือในส่วนที่เป็นโปรแกรม เราอาจจะเขียนโปรแกรม ให้สามารถจดจำเส้นทางที่เคยเดินมาแล้ว พร้อมกับให้มีค่าความน่าจะเป็นที่ไปทางนั้นน้อยลง ซึ่งจะทำให้มีโอกาสที่จะเดินไปทางเส้นทางอื่น เพื่อค้นหาผู้ประสบภัยได้มากยิ่งขึ้น
2. ในส่วนของตัวหุ่นยนต์ อาจจะมีการติดตั้งระบบเซ็นเซอร์ ใช้สำหรับการตรวจสอบสัญญาณชีวิตของผู้ประสบภัย เช่น คาร์บอนไดออกไซด์ ความร้อน เสียง พร้อมกับการติดตั้งกล้องหลายตำแหน่ง เพื่อช่วยในการตรวจสอบหาตำแหน่งของผู้ประสบภัย และมีระบบไฟที่สามารถส่องสว่างได้ในที่มืด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- ฉัตรชัย ฉัตรเจริญสกุล, 2554. การพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการจัดตารางการผลิตแบบตามสั่ง. ปริญญานิพนธ์ปริญญาบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- นิรันดร์ ประวิทย์ธนา, 2545. เก่ง C# ให้ครบสูตร. กรุงเทพฯ : บริษัทวิตตี้ กรุ๊ป จำกัด.
- บัญชา ปะสีละเตสัง, 2552. พัฒนาแอปพลิเคชั่นด้วย Visual C# 2008. กรุงเทพฯ : ซีเอ็ดยูเคชั่น.
- ยุทธนา ลีลาศวัฒนกุล, 2545. คู่มือการเขียนโปรแกรม Visual C# .NET ฉบับวินโดวส์ฟอร์ม. กรุงเทพฯ : หจก.ไทยเจริญการพิมพ์.
- ฤดี มาสุจันท์, 2554. การควบคุมคุณภาพ. กรุงเทพฯ : มีน เซอร์วิส ซัพพลาย.
- สัจจะ จรัสรุ่งรวีวร, 2550, คู่มือ Visual C# 2005 ฉบับสมบูรณ์. กรุงเทพฯ : ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์.
- สัจจะ จรัสรุ่งรวีวร, 2552. เริ่มต้น Visual C# 2008 ฉบับสมบูรณ์. นนทบุรี : บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด.
- สุธี พงศาสกุลชัย, 2550. คัมภีร์ Visual C# 2005. กรุงเทพฯ : เคทีพีคอมพ์แอนด์คอนซัลท์.
- สายชล สินสมบูรณ์ทอง, 2553. สถิติวิศวกรรม. กรุงเทพฯ : จามจุรีโปรดักส์.
- Freedman, 1970. Markov Chains. University of California Bakeley.
- M. Ross, 2007. Introduction to Probability. Models University of California.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
วิธีการใช้งานโปรแกรมการจำลองการค้นหาเส้นทางเดินของหุ่นยนต์

1. การใช้งานโปรแกรม

ส่วนนี้จะเป็นการแสดงขั้นตอนการใช้งานโปรแกรมการจำลองการค้นหาเส้นทางเดินของหุ่นยนต์ โดยเริ่มตั้งแต่การใส่ข้อมูล การประมวลผล และการแสดงผล ซึ่งจะแบ่งขั้นตอนการใช้โปรแกรมออกเป็น 2 ส่วนหลัก ดังนี้

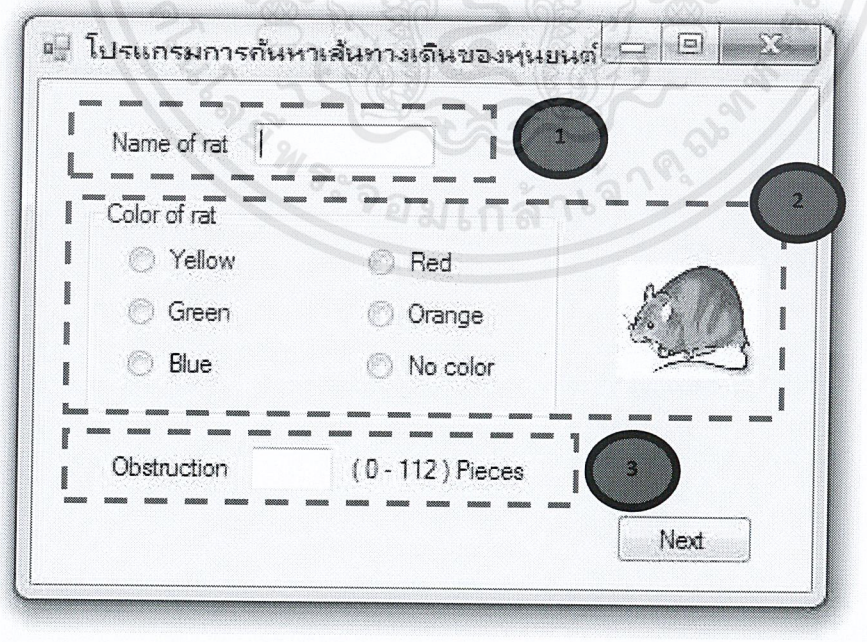
1.1 ขั้นตอนการใช้โปรแกรมในฟอร์ม 1

1.2 ขั้นตอนการใช้โปรแกรมในฟอร์ม 2

1.1 ขั้นตอนการใช้โปรแกรมในฟอร์ม 1

1.1.1 ส่วนรับข้อมูลฟอร์ม 1

เมื่อทำการเปิดโปรแกรม จะมีหน้าจอหลักของโปรแกรมปรากฏขึ้นมา ซึ่งมีส่วนรับข้อมูล ทั้งชื่อ สี และจำนวนสิ่งกีดขวาง ดังรูปที่ ผก 1



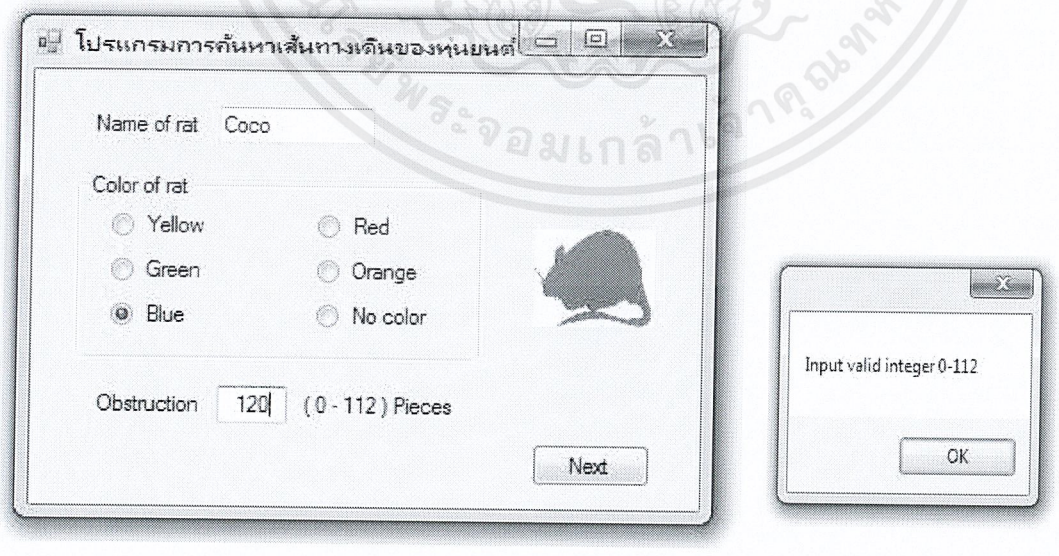
รูปที่ ผก 1 หน้าจอหลักของโปรแกรมการค้นหาเส้นทางเดินของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.2 ส่วนรับข้อมูลฟอร์ม 2

หลังจากที่ทำการใส่ข้อมูลทั้งหมดข้างต้นเรียบร้อยแล้ว ผู้ใช้จะต้องทำการกดปุ่ม Next เพื่อที่จะไปยังฟอร์มที่ 2 ซึ่งก่อนที่ฟอร์มที่ 2 จะปรากฏขึ้นมานั้น จะมีการตรวจสอบค่าที่ผู้ใช้ได้ใส่ในช่องของจำนวนสิ่งกีดขวาง (Obstruction) หากผู้ใช้ลืมใส่ตัวเลขในช่องของจำนวนสิ่งกีดขวาง หรือทำการใส่เป็นตัวหนังสือ โปรแกรม จะทำการแสดงกล่องข้อความขึ้นเตือน ให้ผู้ใช้ใส่เฉพาะตัวเลขเท่านั้น (Input integer only) ดังรูปที่ ผก 2 แต่ถ้าผู้ใช้ใส่ตัวเลขที่มีค่าน้อยกว่าหรือมากกว่าที่กำหนดไว้ โปรแกรมจะแสดงกล่องข้อความขึ้นเตือนให้ผู้ใช้ใส่ตัวเลขได้ไม่เกิน 0-112 (Input valid integer 0-112) ดังรูปที่ ผก 3

รูปที่ ผก 2 แสดงกล่องข้อความขึ้นเตือน ให้ผู้ใช้ใส่เฉพาะตัวเลขเท่านั้น



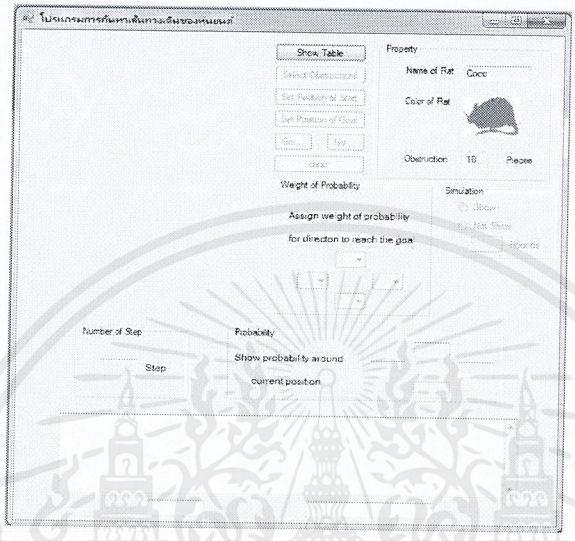
รูปที่ ผก 3 แสดงกล่องข้อความขึ้นเตือน ให้ผู้ใช้ใส่ตัวเลขได้ไม่เกิน 0-112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ขั้นตอนการใช้โปรแกรมในฟอร์ม 2

1.2.1 การแสดงฟอร์ม 2

เมื่อใส่ข้อมูลเสร็จเรียบร้อย ให้ผู้ใช้กดปุ่ม NEXT จากนั้นฟอร์ม 2 จะปรากฏขึ้นมา ดังรูปที่ ผก 4



รูปที่ ผก 4 แสดงฟอร์ม 2

1.2.2 การแสดงตาราง

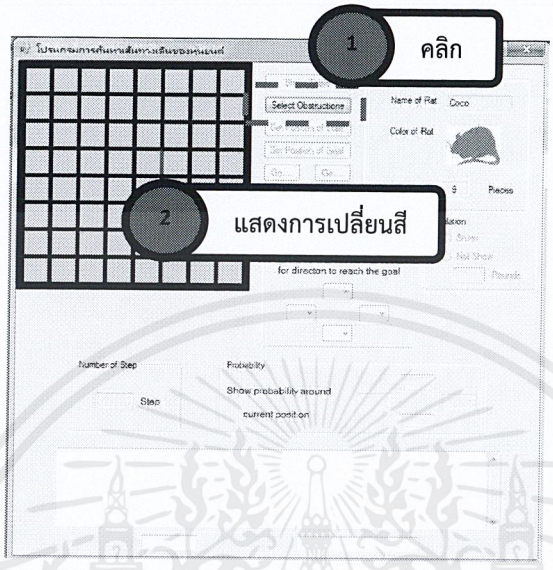
ผู้ใช้จะต้องทำการกดปุ่ม Show Table ก่อน แล้วจะมีตารางขนาด 8x8 แสดงขึ้นมา ดังรูปที่ ผก 5

รูปที่ ผก 5 แสดงตารางขนาด 8x8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.3 การเลือกตำแหน่งสิ่งกีดขวาง

จากนั้นผู้ใช้จะสามารถกดปุ่ม Select Obstruction เพื่อเลือกตำแหน่งสิ่งกีดขวางตามจำนวนที่ผู้ใช้ได้เลือกไว้ในฟอร์มที่ 1 โดยตำแหน่งใดๆที่ถูกเลือก ตำแหน่งนั้นจะเปลี่ยนเป็นสีแดง ดังรูปที่ ผก 6



รูปที่ ผก 6 แสดงการเลือกตำแหน่งสิ่งกีดขวาง

1.2.4 การเลือกตำแหน่งเริ่มต้น

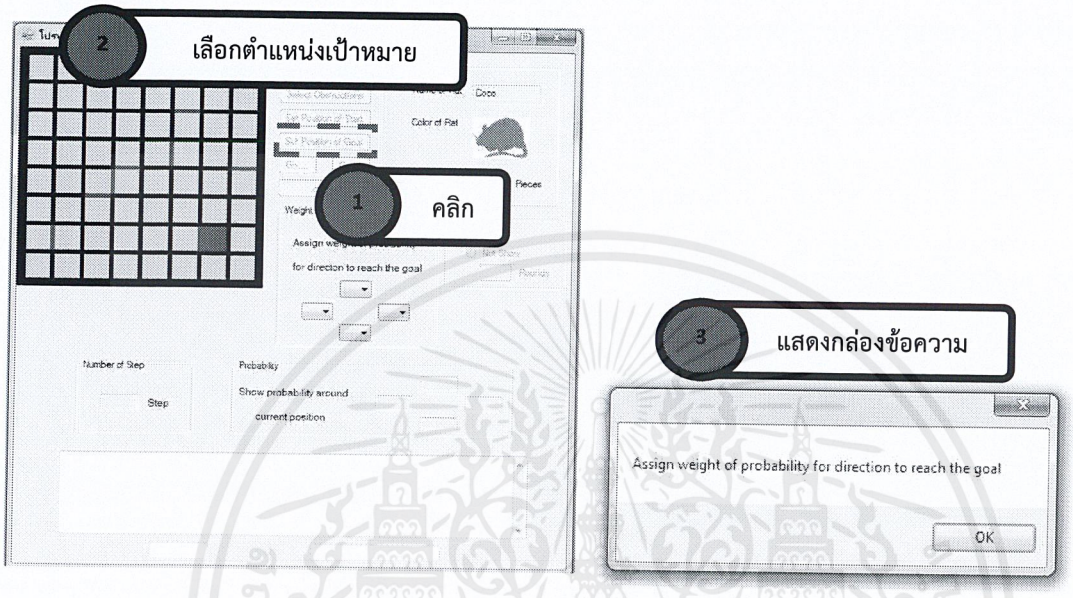
เมื่อผู้ใช้ทำการเลือกตำแหน่งของสิ่งกีดขวางเสร็จเรียบร้อยแล้ว ปุ่ม Set Position of Start จะปรากฏขึ้นมา ผู้ใช้จะต้องกดที่ปุ่มนี้ และทำการเลือกตำแหน่งเริ่มต้น ดังรูปที่ ผก 7

รูปที่ ผก 7 แสดงการเลือกตำแหน่งเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.5 การเลือกตำแหน่งเป้าหมาย

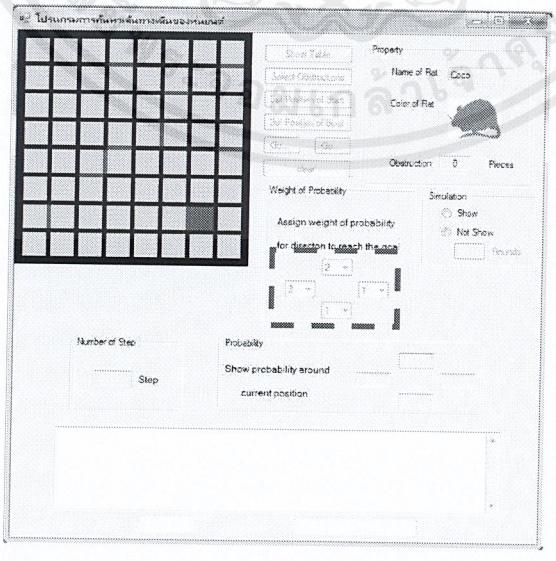
เมื่อผู้ใช้ทำการเลือกตำแหน่งเริ่มต้นเรียบร้อยแล้ว ปุ่ม Set Position of goal จะปรากฏขึ้นมา ผู้ใช้จะต้องกดที่ปุ่มนี้ และทำการเลือกตำแหน่งเป้าหมาย หลังจากนั้นจะมีกล่องข้อความแสดงขึ้นมา เพื่อให้ผู้ใช้ทำการใส่น้ำหนักให้กับค่าความน่าจะเป็นในแต่ละด้าน ดังรูปที่ ผก 8



รูปที่ ผก 8 แสดงการเลือกตำแหน่งเป้าหมาย

1.2.6 การใส่ค่าน้ำหนักของความน่าจะเป็น

หลังจากที่ผู้ใช้ได้เลือกตำแหน่งเริ่มต้น และตำแหน่งเป้าหมายแล้ว ผู้ใช้จะรู้แนวโน้มทิศทางที่จะไปยังเป้าหมาย ทำให้ผู้ใช้สามารถเลือกได้ว่าจะให้ทิศทางใดมีค่าน้ำหนักของความน่าจะเป็นมากกว่า ดังรูปที่ ผก 9



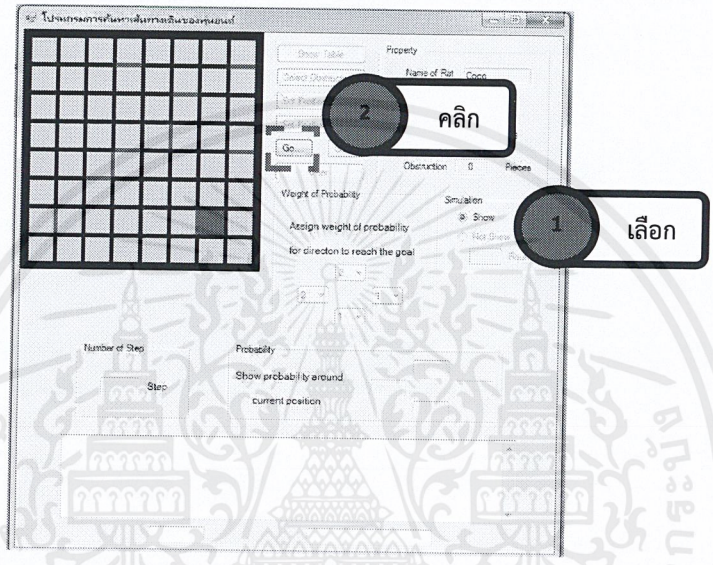
รูปที่ ผก 9 แสดงการใส่น้ำหนักให้กับค่าความน่าจะเป็นในแต่ละด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

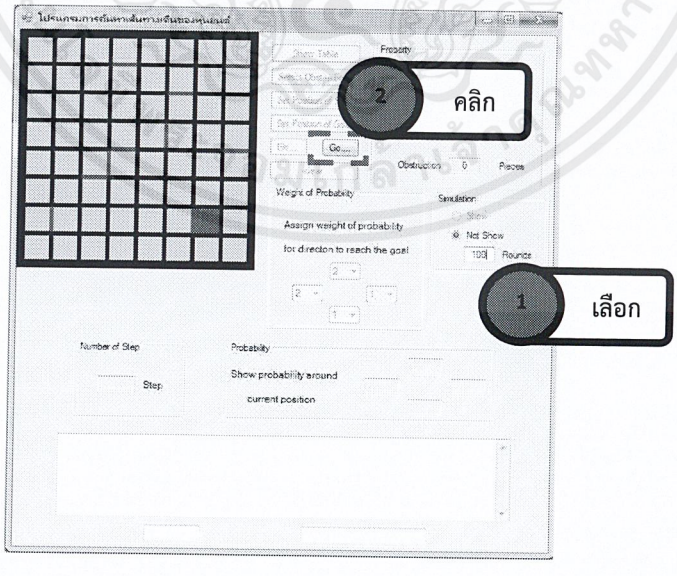
1.2.7 การแสดงการเคลื่อนที่

เมื่อผู้ใช้ทำการใส่ข้อมูลต่างๆเรียบร้อยแล้ว ต่อไปจะเป็นส่วนของ Simulation ซึ่งจะแบ่งเป็น 2 ส่วน ให้ผู้ใช้เลือกว่าจะแสดงการเคลื่อนที่หรือไม่ ดังนี้

- ผู้ใช้เลือกแสดงการเคลื่อนที่ ก็จะสามารถกดปุ่ม Go เพื่อให้โปรแกรมเริ่มประมวลผล ดังรูปที่ ผก 10
- ผู้ใช้เลือกไม่แสดงการเคลื่อนที่ จะมีช่องให้ผู้ใช้ใส่จำนวนรอบในการเดิน เมื่อผู้ใช้ใส่จำนวนรอบในการเดินแล้ว ก็จะสามารถกด Go ดังรูปที่ ผก 11 เพื่อให้โปรแกรมเริ่มประมวลผลได้



รูปที่ ผก 10 แสดงการเลือกแสดงการเคลื่อนในส่วนของ Simulation



รูปที่ ผก 11 แสดงการเลือกไม่แสดงการเคลื่อนที่ในส่วนของ Simulation

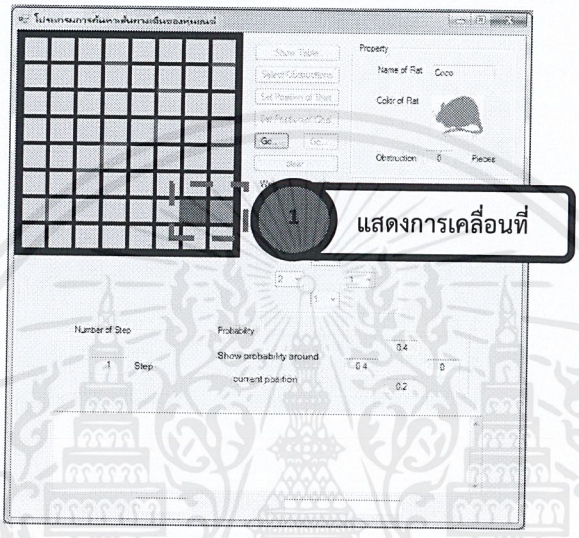
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.8 การแสดงผล

หลังจากที่โปรแกรมทำการประมวลผล ต่อไปจะเป็นการแสดงผล ซึ่งจะแบ่งเป็น 2 ส่วน ดังนี้

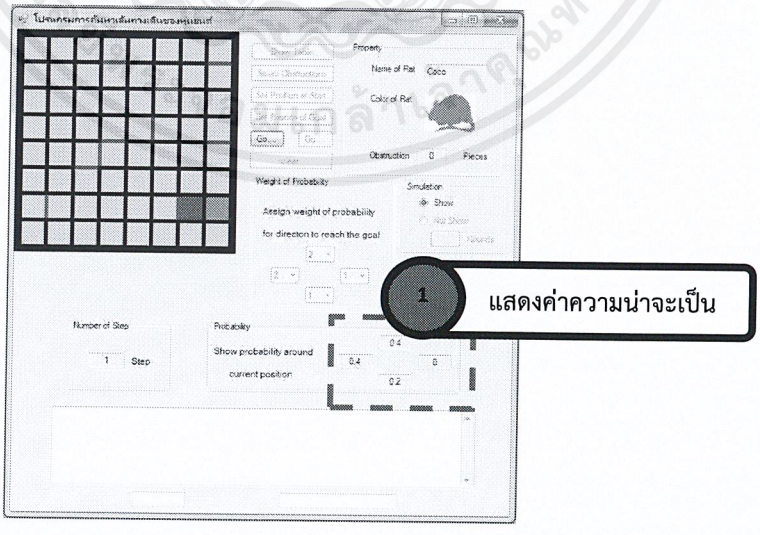
- หากผู้ใช้เลือกแสดงการเคลื่อนที่ เมื่อกดปุ่ม Go โปรแกรมจะทำการประมวลผล และแสดงผลลัพธ์ ดังนี้

- การเคลื่อนที่จากตำแหน่งเริ่มต้นไปยังตำแหน่งเป้าหมาย ในระหว่างการเคลื่อนที่ ตำแหน่งที่เป็นตำแหน่งปัจจุบันจะถูกเปลี่ยนให้เป็นสีน้ำตาล และเมื่อเคลื่อนที่ไปยังตำแหน่งต่อไป ตำแหน่งก่อนหน้าจะถูกเปลี่ยนให้เป็นสีฟ้าดังเดิม ดังรูปที่ ผก 12



รูปที่ ผก 12 แสดงการเคลื่อนที่

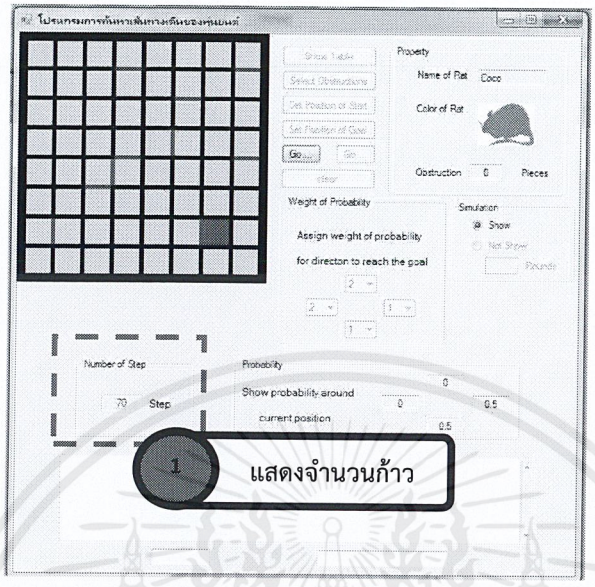
- แสดงค่าความน่าจะเป็นของแต่ละทิศทางรอบๆตำแหน่งปัจจุบัน ดังรูปที่ ผก 13



รูปที่ ผก 13 แสดงค่าความน่าจะเป็นของแต่ละทิศทางรอบๆตำแหน่งปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

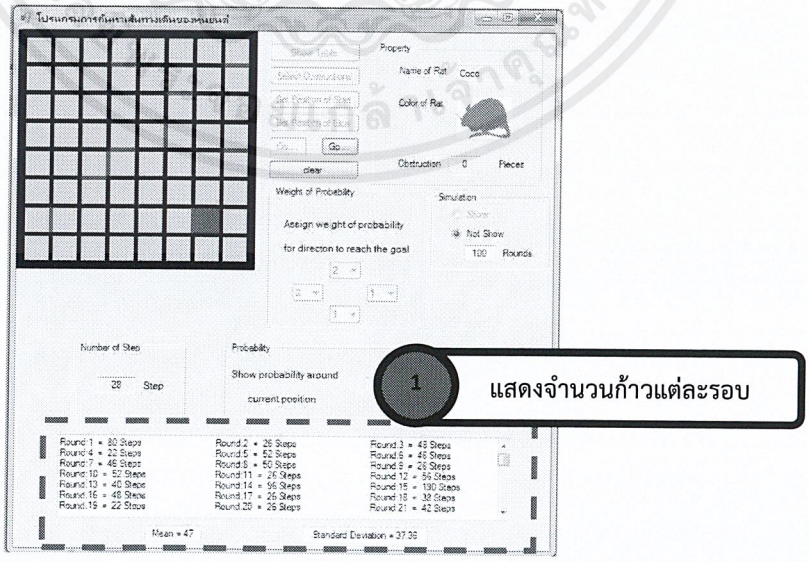
- ในระหว่างการเคลื่อนที่ จะมีการแสดงจำนวนก้าว ตั้งแต่ก้าวแรกจนถึงก้าวสุดท้าย ดังรูปที่ ผก 14



รูปที่ ผก 14 แสดงจำนวนก้าว ตั้งแต่ก้าวแรกจนถึงก้าวสุดท้าย

- หากผู้ใช้เลือกไม่แสดงการเคลื่อนที่ หลังจากที่ใส่จำนวนรอบในการเดิน และกดปุ่ม Go โปรแกรมจะทำการประมวลผล และแสดงผลลัพธ์ โดยจะมีการแสดงผลลัพธ์ที่เหมือนกับการเลือกแสดงการเคลื่อนที่ คือ ค่าความน่าจะเป็นของแต่ละทิศทางรอบๆตำแหน่งปัจจุบัน และจำนวนก้าว ตั้งแต่ก้าวแรกจนถึงก้าวสุดท้าย ส่วนผลลัพธ์ที่แตกต่างกัน มีดังนี้

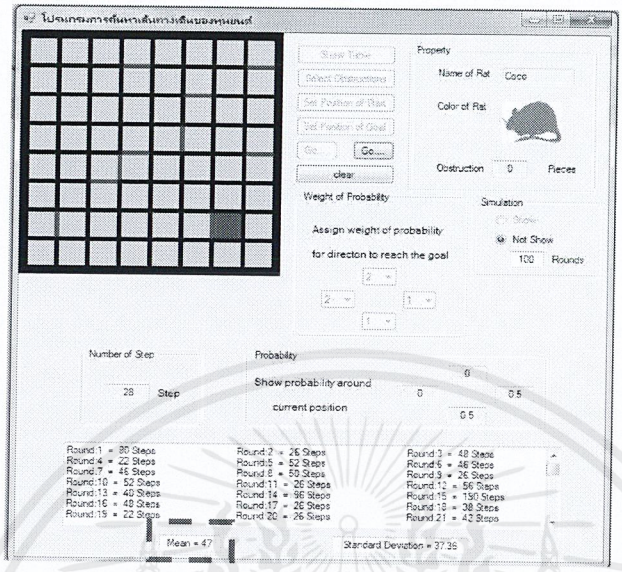
- แสดงจำนวนก้าวที่เกิดขึ้นในแต่ละรอบของการเดิน ดังรูปที่ ผก 15



รูปที่ ผก 15 แสดงจำนวนก้าวที่เกิดขึ้นในแต่ละรอบของการเดิน

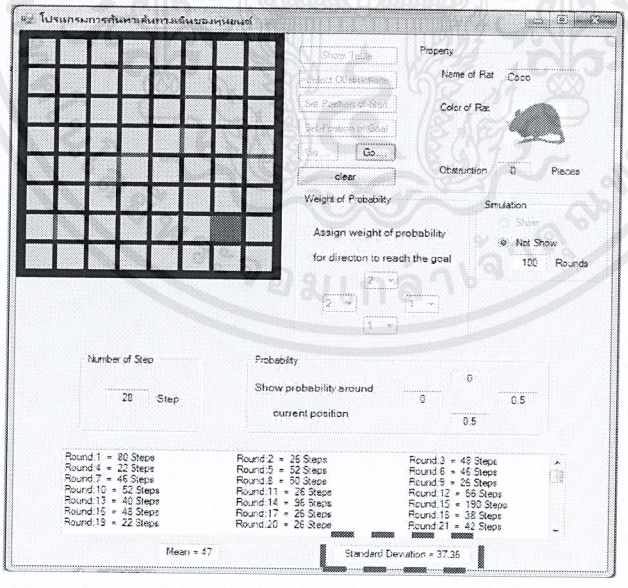
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แสดงค่าเฉลี่ยของจำนวนก้าว ซึ่งคำนวณจากวิธีการคำนวณค่าเฉลี่ยตัวอย่าง ดังรูปที่ ผก 16



รูปที่ ผก 16 แสดงค่าเฉลี่ยของจำนวนก้าว

- แสดงค่าส่วนเบี่ยงเบนมาตรฐานของจำนวนก้าว คำนวณได้จากวิธีการคำนวณค่าส่วนเบี่ยงเบนมาตรฐานตัวอย่าง ดังรูปที่ ผก 17



รูปที่ ผก 17 แสดงค่าส่วนเบี่ยงเบนมาตรฐานของจำนวนก้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
ตัวอย่างการเขียนโปรแกรมการจำลองการค้นหาเส้นทางเดินของหุ่นยนต์

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาและพัฒนาโปรแกรมคอมพิวเตอร์ เพื่อใช้ในการจำลองการค้นหาเส้นทางเดินของหุ่นยนต์ โดยใช้กระบวนการตัดสินใจมาร์คอฟ ซึ่งในโปรแกรมมีหน้าต่างที่ 1 หรือฟอร์มที่ 1 ใช้ในการรับข้อมูล และมีหน้าต่างที่ 2 หรือฟอร์มที่ 2 ใช้ในการรับข้อมูลสำหรับการประมวลผล และใช้ในการแสดงผลจากการประมวล ตัวอย่างการเขียนโปรแกรมการจำลองการค้นหาเส้นทางเดินของหุ่นยนต์ มีดังนี้

1. ฟอร์มที่ 1

ในฟอร์มที่ 1 จะเป็นการเขียนโค้ดเพื่อรับข้อมูลทั้งชื่อ สี และจำนวนสิ่งกีดขวาง และมีการส่งข้อมูลทั้งหมดเพื่อไปแสดงในฟอร์มที่ 2 โดยมีโค้ดดังนี้

```
public partial class Form1 : Form
    {
      public delegate void SendDataToForm2(string txt);
      public delegate void SendCountToForm2(string count);
      public int color_case = 0;
      public int counter_pb = 0;
      public bool checkName = true;

      public Form1()
        {
            InitializeComponent();
        }

     private void radioButton1_CheckedChanged(object sender,
EventArgs e)
        {
            color_case = 5;
            pictureBox1.Visible = false;
            pictureBox2.Visible = true;
            pictureBox3.Visible = false;
            pictureBox4.Visible = false;
            pictureBox5.Visible = false;
            pictureBox6.Visible = false;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      private void radioButton2_CheckedChanged(object sender,
 EventArgs e)
        {
            color_case = 1;
            pictureBox1.Visible = false;
            pictureBox2.Visible = false;
            pictureBox3.Visible = true;
            pictureBox4.Visible = false;
            pictureBox5.Visible = false;
            pictureBox6.Visible = false;
        }

      private void radioButton6_CheckedChanged(object sender,
 EventArgs e)
        {
            color_case = 0;
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            pictureBox4.Visible = false;
            pictureBox5.Visible = false;
            pictureBox6.Visible = false;
            pictureBox1.Visible = true;
        }

      private void radioButton3_CheckedChanged(object sender,
 EventArgs e)
        {
            color_case = 2;
            pictureBox1.Visible = false;
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            pictureBox4.Visible = true;
            pictureBox5.Visible = false;
            pictureBox6.Visible = false;
        }

      private void radioButton4_CheckedChanged(object sender,
 EventArgs e)
        {
            color_case = 3;
            pictureBox1.Visible = false;
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            pictureBox4.Visible = false;
            pictureBox5.Visible = true;
            pictureBox6.Visible = false;
        }

      private void radioButton5_CheckedChanged(object sender,
 EventArgs e)
        {
            color_case = 4;
            pictureBox1.Visible = false;
            pictureBox2.Visible = false;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pictureBox3.Visible = false;
            pictureBox4.Visible = false;
            pictureBox5.Visible = false;
            pictureBox6.Visible = true;
        }

      private void button1_Click(object sender, EventArgs e)
        {
            String input = this.textBox2.Text;
            try
            {
                  int check = Convert.ToInt32(input);
                  if (check >= 0 && check < 113)
                  {
                     counter_pb = check;
                     Form2 page2 = new Form2();
                     SendDataToForm2 mydelegate = new SendDataToForm2
(page2.GetData);
                     mydelegate(this.textBox1.Text);
                     SendCountToForm2 mycount = new SendCountToForm2
(page2.Getcount);
                     mycount(this.textBox2.Text);

                     page2.Show();
                  }
                  else
                  {
                     counter_pb = 0;
                     MessageBox.Show("Input valid integer 0-112");
                  }
            }
            catch
            {
                MessageBox.Show("Input integer only");
            }
        }
    }
```

2. ฟอร์มที่ 1

ในฟอร์มที่ 2 จะเป็นการเขียนโค้ดเพื่อรับข้อมูลทั้งตำแหน่งของสิ่งกีดขวาง ตำแหน่งเริ่มต้น ตำแหน่งเป้าหมาย และค่าน้ำหนักของความน่าจะเป็น นอกจากนั้นหากผู้ใช้ต้องการให้เดินหลายรอบ จะมีการรับค่าจำนวนรอบในการเดิน เมื่อรับค่าเสร็จแล้ว จะมีการประมวลผล และแสดงผลจากการประมวลผล โดยมีโค้ดดังนี้

```
public partial class Form2 : Form
    {
      public MyPanel myPanel;
      private SubPanel[,] panel_output;
      private SubPanel[,] obrow_output;
      private SubPanel[,] obcol_output;
      private SubPanel currentSlot = null;
      private SubPanel background_output;
      public string getcount_str;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public string getcount_str;
public int getcount_num;
private bool enableObstruction = false;
private bool isSetTarget = false;
private bool isSetStart = false;
private bool beStartRun = false;
private bool Deletecolor = false;
private int indexFirstPoint = -1;
private delegate void UpdateTextBox(object item);
public double randomOUT;
public int countT, countB, countL, countR;
public int wT,wB,wL,wR;
public int wTChange=0;
public int wBChange = 0;
public int wLChange = 0;
public int wRChange = 0;
public int CountStep = 0;
private int finalIndexPoint = 0;
public bool gofT, gofB, gofL, gofR = false;
publicint ForStart = 0;
public int ForShow = 0;
public int ForNotShow = 0;
public int sec = 500;
public int roundCtl = 0;
public int p = 0;
public int[] countSet;
public int[] Std;
public bool isSpeedMode = false;
public int r = 1;
public bool isInitWeight = false;
public bool NextRound = false;
Thread thread;

public Form2()
  {
      InitializeComponent();
      myPanel = new MyPanel(9, 8, 8, 9, 8, 8);
      panel_output = myPanel.getBlockPanel();
      obrow_output = myPanel.getPanelRow();
      obcol_output = myPanel.getPanelCol();
      background_output = myPanel.getBackGround();
      button2.Enabled = false;
      button3.Enabled = false;
      button4.Enabled = false;
      button5.Enabled = false;
      button6.Enabled = false;
      button7.Enabled = false;
      wTComboBox.Enabled = false;
      wBComboBox.Enabled = false;
      wLComboBox.Enabled = false;
      wRComboBox.Enabled = false;
      radioButton1.Enabled = false;
      radioButton2.Enabled = false;
      textBox3.Enabled = false;
      label8.Enabled = false;
      randomOUT = 0.00d;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            countT = 0;
            countB = 0;
            countL = 0;
            countR = 0;
            wT = 0;
            wB = 0;
            wL = 0;
            wR = 0;
            button6.Enabled = false;

            switch (Program.firstForm.color_case)
            {
                case 0:
                    pictureBox1.Visible = true;
                    break;
                case 1:
                    pictureBox3.Visible = true;
                    break;
                case 2:
                    pictureBox4.Visible = true;
                    break;
                case 3:
                    pictureBox5.Visible = true;
                    break;
                case 4:
                    pictureBox6.Visible = true;
                    break;
                default:
                    pictureBox2.Visible = true;
                    break;
            }
        }
    public void button1_Click_1(object sender, EventArgs e)
        {
            button2.Enabled = true;
            button1.Enabled = false;
            int count_panel = 0;
            int count_col = 0;
            int count_row = 0;

            for (int j = 0; j < 8; j++)
            {

                for (int i = 0; i < 8; i++)
                {
                    panel_output[i, j] = new SubPanel();
                    panel_output[i, j].setPosition(j, i);
                    panel_output[i, j].index = count_panel;
                    count_panel += 1;
                    panel_output[i, j].Location = new
System.Drawing.Point(10 + i * 35, 10 + j * 35);
                    panel_output[i, j].Size = new
System.Drawing.Size(30,30);
                    panel_output[i, j].BackColor = Color.LightBlue;
                    panel_output[i, j].Click += new EventHandler
(panelSlot_Click);
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                this.Controls.Add(panel_output[i, j]);
            }
        }

        for (int j = 0; j < 9; j++)
        {
            for (int i = 0; i < 8; i++)
            {
                obrow_output[i, j] = new SubPanel();
                obrow_output[i, j].setPosition(j, i);
                obrow_output[i, j].index = count_row;
                obrow_output[i, j].isRowOpstruct = true;
                count_row += 1;
                obrow_output[i,j].Location = new
System.Drawing.Point(10 + i * 35, 5 + j * 35);
                obrow_output[i, j].Size = new
System.Drawing.Size(30,5);
                obrow_output[i, j].BackColor = Color.Black;
                obrow_output[i, j].Click += new
EventHandler(obstruct_Click);
                this.Controls.Add(obrow_output[i, j]);
            }
        }

        for (int j = 0; j < 8; j++)
        {

            for (int i = 0; i < 9; i++)
            {
                obcol_output[i, j] = new SubPanel();
                obcol_output[i, j].setPosition(j, i);
                obcol_output[i, j].index = count_col;
                obcol_output[i, j].isColOpstruct=true;
                count_col += 1;
                obcol_output[i, j].Location = new
System.Drawing.Point(5 + i * 35, 10 + j * 35);
                obcol_output[i, j].Size = new System.Drawing.Size
(5, 30);
                obcol_output[i, j].BackColor = Color.Black;
                obcol_output[i, j].Click += new EventHandler
(obstruct_Click);
                this.Controls.Add(obcol_output[i, j]);
            }
        }

        background_output.Location = new System.Drawing.Point(0, 0);
        background_output.Size = new System.Drawing.Size(295, 295);
        background_output.BackColor = Color.Black;
        this.Controls.Add(background_output);
    }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // Search position
        private SubPanel searchPanelArrayByIndex(int idex, SubPanel[,]
 inputP)
        {
            SubPanel res = null;
            foreach (SubPanel obr in inputP)
            {
               if(obr.index == idex){
                   res = obr;
                   break;
               }
            }
            return res;
        }

        private SubPanel searchPanelArrayByCxCy(int Cx, int Cy,
 SubPanel[,] inputP)
        {
            SubPanel res = null;
            foreach (SubPanel obr in inputP)
            {
                if ((obr.getPosition().X == Cx) && (obr.getPosition().Y
 == Cy))
                {
                    res = obr;
                    break;
                }
            }
            return res;
        }

        // get Next panel by Direction.
        private SubPanel getNextStep(String direc, int currentIndex)
        {
            SubPanel outPa = null;
            if(direc.Equals("UP") && (currentIndex - 8 >= 0)){
                foreach (SubPanel pa2 in panel_output)
                {
                    if(pa2.index == currentIndex - 8)
                    {
                        outPa = pa2;
                    }
                }
            }else if(direc.Equals("DOWN") && (currentIndex +8 < 64)){
                foreach (SubPanel pa2 in panel_output)
                {
                    if (pa2.index == currentIndex + 8)
                    {
                        outPa = pa2;
                    }
                }
            }
            else if (direc.Equals("LEFT") && (currentIndex != 0 &&
currentIndex != 8 && currentIndex != 16 && currentIndex != 24 &&
currentIndex != 32 && currentIndex != 40 && currentIndex != 48 &&
currentIndex != 56))
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                foreach (SubPanel pa2 in panel_output)
                {
                    if (pa2.index == currentIndex - 1)
                    {
                        outPa = pa2;
                    }
                }

            }else{if (currentIndex != 7 && currentIndex != 15 &&
currentIndex != 23 && currentIndex != 31 && currentIndex != 39 &&
currentIndex != 47 && currentIndex != 55 && currentIndex != 63 )
                 {
                    foreach (SubPanel pa2 in panel_output)
                    {
                        if (pa2.index == currentIndex + 1)
                        {
                            outPa = pa2;
                        }
                    }
                 }
            }
            return outPa;
        }

        // keep probability
        private void panelSlot_Click(object sender, EventArgs e)
        {
            SubPanel pa = (SubPanel)sender;
            SubPanel outcomeTop;
            SubPanel outcomeBottom;
            SubPanel outcomeLeft;
            SubPanel outcomeRight;

            // Set prob of border = 0
            if (pa.index == 0 || pa.index <= 7)
            {
                outcomeTop = searchPanelArrayByIndex(pa.index,
obrow_output);
                pa.ObrT = 0;
                pa.decreaseWeightRemainder();
            }
            if (pa.index == 0 || pa.index == 8 || pa.index == 16 ||
pa.index == 24 || pa.index == 32 || pa.index == 40 || pa.index == 48 ||
pa.index == 56)
            {
                outcomeLeft = searchPanelArrayByCxCy(pa.getPosition().X,
pa.getPosition().Y, obcol_output);
                pa.ObrL = 0;
                pa.decreaseWeightRemainder();
            }
            if (pa.index == 7 || pa.index == 15 || pa.index == 23 ||
pa.index == 31 || pa.index == 39 || pa.index == 47 || pa.index == 55 ||
pa.index == 63)
            {
                outcomeRight =
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
searchPanelArrayByCxCy(pa.getPosition().X, pa.getPosition().Y + 1,
obcol_output);
                pa.ObrR = 0;
                pa.decreaseWeightRemainder();
            }
            if (pa.index == 56 || (pa.index <= 63 && pa.index >= 56))
            {
                outcomeBottom = searchPanelArrayByIndex(pa.index + 8,
obrow_output);
                pa.ObrB = 0;
                pa.decreaseWeightRemainder();
            }

            //Set Start
            if (isSetStart)
            {
                pa.BackColor = Color.Green;
                pa.panel_type = 1;
                pa.isStart = true;
                isSetStart = false;
                button3.Enabled = false;
                button4.Enabled = true;
            }

            //Set Goal
            if (isSetTarget && !pa.isStart)
            {
                pa.BackColor = Color.Gold;
                pa.panel_type = 2;
                pa.isTarget = true;
                isSetTarget = false;
                button4.Enabled = false;
                MessageBox.Show("Assign weight of probability for
direction to reach the goal");
            }
            if (wTChange == 1 && wBChange == 1 && wLChange == 1 &&
wRChange == 1)
            {
                wT =
Convert.ToInt32(wTComboBox.SelectedItem.ToString());
                wB =
Convert.ToInt32(wBComboBox.SelectedItem.ToString());
                wL =
Convert.ToInt32(wLComboBox.SelectedItem.ToString());
                wR =
Convert.ToInt32(wRComboBox.SelectedItem.ToString());
            }

            //panel Slot Click
            if (!pa.getFootPrint())
            {
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Survey Obstruction around
            //Check around (top)
            outcomeTop = searchPanelArrayByIndex(pa.index, obrow_output);
                 if (outcomeTop.getFootPrint())
                 {
                     pa.ObrT = 0;
                     pa.decreaseWeightRemainder();
                 }

            //Check around (bottom)
            outcomeBottom = searchPanelArrayByIndex(pa.index + 8,
obrow_output);
                 if (outcomeBottom.getFootPrint())
                 {
                     pa.ObrB = 0;
                     pa.decreaseWeightRemainder();
                 }

            //Check around (left)
            outcomeLeft = searchPanelArrayByCxCy(pa.getPosition().X,
pa.getPosition().Y, obcol_output);
                 if (outcomeLeft.getFootPrint())
                 {
                     pa.ObrL = 0;
                     pa.decreaseWeightRemainder();
                 }

            //Check around (right)
            outcomeRight = searchPanelArrayByCxCy(pa.getPosition().X,
pa.getPosition().Y + 1, obcol_output);
                 if (outcomeRight.getFootPrint())
                 {
                     pa.ObrR = 0;
                     pa.decreaseWeightRemainder();
                 }

            //Assign prob
            float SumProbObr = 0.00f;
            SumProbObr = (pa.ObrT * wT) + (pa.ObrB * wB) + (pa.ObrL *
wL)+(pa.ObrR * wR);
                 if (SumProbObr != 0)
                 {
                     pa.probT = (pa.ObrT * wT) / SumProbObr;
                     pa.probB = (pa.ObrB * wB) / SumProbObr;
                     pa.probL = (pa.ObrL * wL) / SumProbObr;
                     pa.probR = (pa.ObrR * wR) / SumProbObr;

                 }
                 else
                 {
                     pa.probT = 0;
                     pa.probB = 0;
                     pa.probL = 0;
                     pa.probR = 0;
                 }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            //Asign prop to cumulative Function
                 pa.inittialCumulativeT();
                 pa.inittialCumulativeB();
                 pa.inittialCumulativeL();
                 pa.inittialCumulativeR();
                 randomOUT = pa.getRandom();

            //scan and get prob
                 if (currentSlot != null)
                 {
                     pa.setFootPrint(true);
                 }
                 }
                 else
                 {
                 pa.resetWeightRemainder();
                 pa.probT = 1;
                 pa.probB = 1;
                 pa.probL = 1;
                 pa.probR = 1;
                 pa.setFootPrint(false);
                 }

             //check range of probability & choose direction
             if (randomOUT >= 0 && randomOUT < pa.getCDFT())
             {
                 pa.stepAction = "UP";
             }
             else if (randomOUT >= pa.getCDFT() && randomOUT <
pa.getCDFB())
             {
                 pa.stepAction = "DOWN";
             }
             else if (randomOUT >= pa.getCDFB() && randomOUT <
pa.getCDFL())
             {
                 pa.stepAction = "LEFT";
             }
             else if (randomOUT >= pa.getCDFL() && randomOUT <
pa.getCDFR())
             {
                 pa.stepAction = "RIGHT";
             }
         }

         private void obstruct_Click(object sender, EventArgs e)
         {
             //for increase&decrease # of obstruction and change color
             SubPanel pa = (SubPanel)sender;
             getcount_num = Convert.ToInt32(textBox2.Text);
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Test condition of borderX
if ((pa.getPosition().X != 0 && pa.getPosition().X != 8))
{
    if (pa.getPosition().Y != 0 && pa.getPosition().Y != 8)
    {
        if ((getcount_num > 0) && (enableObstruction))
        {
            if (pa.getFootPrint
            {
                if (pa.BackColor == Color.Black)
                {
                    pa.BackColor = Color.Red;
                    getcount_num--;
                    textBox2.Text = getcount_num.ToString();
                    pa.setFootPrint(true);
                }
                else
                {
                    pa.BackColor = Color.Black;
                    getcount_num++;
                    textBox2.Text = getcount_num.ToString();
                    pa.setFootPrint(false);
                }
            }
            else
            {
                pa.BackColor = Color.Red;
                pa.setFootPrint(true);
                getcount_num--;
                textBox2.Text = getcount_num.ToString();
            }

        }
    }
    else if (pa.getPosition().Y==0 && pa.isRowOpstruct)
    {
        if ((getcount_num > 0) && (enableObstruction))
        {


            if (pa.getFootPrint()                                {
                if (pa.BackColor == Color.Black)
                {
                    pa.BackColor = Color.Red;
                    getcount_num--;
                    textBox2.Text = getcount_num.ToString();
                    pa.setFootPrint(true);
                }

                else
                {
                    pa.BackColor = Color.Black;
                    getcount_num++;
                    textBox2.Text = getcount_num.ToString();
                    pa.setFootPrint(false);
                }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                        else
                        {
                            pa.BackColor = Color.Red;
                            pa.setFootPrint(true);
                            getcount_num--;
                            textBox2.Text = getcount_num.ToString();
                        }
                    }
                }
            }
            else if (!pa.isRowOpstruct && pa.getPosition().Y != 0 &&
pa.getPosition().Y!=8)
            {
                if ((getcount_num > 0) && (enableObstruction))
                {

                    if (pa.getFootPrint()                    {
                        if (pa.BackColor == Color.Black)
                        {
                            pa.BackColor = Color.Red;
                            getcount_num--;
                            textBox2.Text = getcount_num.ToString();
                            pa.setFootPrint(true);
                        }
                        else
                        {
                            pa.BackColor = Color.Black;
                            getcount_num++;
                            textBox2.Text = getcount_num.ToString();
                            pa.setFootPrint(false);
                        }
                    }
                    else
                    {
                        pa.BackColor = Color.Red;
                        pa.setFootPrint(true);
                        getcount_num--;
                        textBox2.Text = getcount_num.ToString();
                    }
                }
            }
            else {}

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            string In = textBox2.Text;
            int Num = Convert.ToInt32(In);
            if (Num == 0)
            {
                button3.Enabled = true;
                button2.Enabled = false;
            }
        }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        public void GetData(string txt)
        {
            textBox1.Text = txt;       // name of rat link from form1
        }
        public void Getcount(string count)
        {
            textBox2.Text = count;    // # of obstruction link fromm
form1
        }

        private void button2_Click_1(object sender, EventArgs e)
        {
            enableObstruction = true;
        }

        private void button4_Click(object sender, EventArgs e)
        {
            isSetTarget = true;
            wTComboBox.Enabled = true;
            wBComboBox.Enabled = true;
            wLComboBox.Enabled = true;
            wRComboBox.Enabled = true;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            isSetStart = true;
        }

        private void button5_Click(object sender, EventArgs e)
        {
            lunch();
            button6.Enabled = false;
        }

        private void lunch()
        {
            routeAlgorithm();
            ///// use for restart buttom Go!!!!!!!!!!!!!!!!!!!!!!!!!!!!
            beStartRun = false;
            currentSlot = null;
            CountStep = 0;
            ////////////////////////////////////////////////////////////
            if (!isInitWeight)
            {
                wT =
Convert.ToInt32(wTComboBox.SelectedItem.ToString());
                wB =
Convert.ToInt32(wBComboBox.SelectedItem.ToString());
                wL =
Convert.ToInt32(wLComboBox.SelectedItem.ToString());
                wR =
Convert.ToInt32(wRComboBox.SelectedItem.ToString());
            }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            isInitWeight = true;
                thread = new Thread(new ThreadStart(toWalk));
                thread.Start();
        }

        public void toWalk()
        {
            int pd=0;
            //Find Terminate point
            //SubPanel compair = null;
            foreach (SubPanel pt in panel_output)
           {
                if (pt.isTarget)
                {
                    pd = pt.index;
                    finalIndexPoint = pd;
                    //compair = pt;
                }
           }

           while ((currentSlot == null || (currentSlot.index != pd))){
               try
               {
                   //Find Start point
                   if (currentSlot == null)
                   {
                       SubPanel gp = null;
                       foreach (SubPanel pa3 in panel_output)
                       {
                           if (pa3.isStart)
                           {
                               gp = pa3;
                               indexFirstPoint = pa3.index;
                           }
                       }
                       currentPosition(gp);
                   }
                   else
                   {

currentPosition(getNextStep(currentSlot.stepAction, currentSlot.index));
                   }
                   Thread.Sleep(sec);  // 1 sec.
                   // for delete color panel//////////////////////////
                   if (currentSlot != null && Deletecolor == true &&
currentSlot.index != indexFirstPoint && currentSlot.index !=
finalIndexPoint) { currentSlot.BackColor = Color.LightBlue; }
               }
               catch (Exception e)
               {
                   MessageBox.Show("System ERROR !!!! " + e);
                   System.Environment.Exit(0);
               }
           }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if( p < roundCtl-1 && isSpeedMode){

                countSet[p] = CountStep;
                p++;

                lunch();
            }
            else if (p == roundCtl-1)
            {
                countSet[p] = CountStep;
                Std = new int[roundCtl];
                String result = "";
                int result2 = 0;
                int result3 = 0;
                int result4 = 0;
                int result5 = 0;

                // run another round
                NextRound = true;
                for (int x = 0; x < roundCtl; x++)
                {
                    result = result + "Round:" + (x + 1) + " " + " = " +
" " + countSet[x] +" Steps"+ "\t" + "\t";
                    result2 = result2 + countSet[x];
                }
                result3 = result2 / roundCtl;
                this.AddPtextBoxItemRS(result.ToString());
                this.AddPtextBoxItemM("Mean = " + result3.ToString());

                // for calculate standard deviation
                for (int s = 0; s < roundCtl; s++)
                {
                    Std[s] = Std[s] + ((countSet[s] -
result3)*(countSet[s] - result3));
                    result4 = result4 + Std[s];
                }
                result5 = result4 / (roundCtl - 1);
                double mySquareRoot = Math.Sqrt(result5);
                float AndStd = 0.00f;
                AndStd = (float)mySquareRoot;
                this.AddPtextBoxItemStd("Standard Deviation = " +
AndStd.ToString("####.##"));
            }

            if (!isSpeedMode){
                this.currentSlot.BackColor = Color.Gold;
                MessageBox.Show("Mission completed!!!!!");
            }
        }

        private void currentPosition(SubPanel incom)
        {
            SubPanel pa = incom;
            SubPanel outcomeTop;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SubPanel outcomeBottom;
            SubPanel outcomeLeft;
            SubPanel outcomeRight;

            // Set prob of border = 0

            if (pa.index == 0 || pa.index <= 7)
            {
                outcomeTop = searchPanelArrayByIndex(pa.index,
obrow_output);
                pa.ObrT = 0;
                pa.decreaseWeightRemainder();
            }
            if (pa.index == 0 || pa.index == 8 || pa.index == 16 ||
pa.index == 24 || pa.index == 32 || pa.index == 40 || pa.index == 48 ||
pa.index == 56)
            {
                outcomeLeft = searchPanelArrayByCxCy(pa.getPosition().X,
pa.getPosition().Y, obcol_output);
                pa.ObrL = 0;
                pa.decreaseWeightRemainder();
            }

            if (pa.index == 7 || pa.index == 15 || pa.index == 23 ||
pa.index == 31 || pa.index == 39 || pa.index == 47 || pa.index == 55 ||
pa.index == 63)
            {
                outcomeRight =
searchPanelArrayByCxCy(pa.getPosition().X, pa.getPosition().Y + 1,
obcol_output);
                pa.ObrR = 0;
                pa.decreaseWeightRemainder();
            }

            if (pa.index == 56 || (pa.index <= 63 && pa.index >= 56))
            {
                outcomeBottom = searchPanelArrayByIndex(pa.index + 8,
obrow_output);
                pa.ObrB = 0;
                pa.decreaseWeightRemainder();
            }

            //Set Start
            if (isSetStart)
            {
                pa.BackColor = Color.Green;
                pa.panel_type = 1; // 1 for start

                // set Dx , Dy
                pa.isStart = true;
                isSetStart = false;
                button3.Enabled = false;
                button4.Enabled = true;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            //Set Goal
            if (isSetTarget && !pa.isStart)
            {
                pa.BackColor = Color.Gold;
                pa.panel_type = 2;
                pa.isTarget = true;
                isSetTarget = false;
                button4.Enabled = false;
            }

 //Survey Obstruction around-----------------------------------------

                //Check around (top)
                outcomeTop = searchPanelArrayByIndex(pa.index,
 obrow_output);
                if (outcomeTop.getFootPrint())
                {
                    pa.ObrT = 0;
                    pa.decreaseWeightRemainder();
                }

                //Check around (bottom)
                outcomeBottom = searchPanelArrayByIndex(pa.index + 8,
 obrow_output);
                if (outcomeBottom.getFootPrint())
                {
                    pa.ObrB = 0;
                    pa.decreaseWeightRemainder();
                }

                //Check around (left)
                outcomeLeft = searchPanelArrayByCxCy(pa.getPosition().X,
 pa.getPosition().Y, obcol_output);
                if (outcomeLeft.getFootPrint())
                {
                    pa.ObrL = 0;
                    pa.decreaseWeightRemainder();
                }

                //Check around (right)
                outcomeRight =
 searchPanelArrayByCxCy(pa.getPosition().X, pa.getPosition().Y + 1,
 obcol_output);
                if (outcomeRight.getFootPrint())
                {
                    pa.ObrR = 0;
                    pa.decreaseWeightRemainder();
                }

                //Assign prob
                float SumProbObr = 0.00f;
                SumProbObr = (pa.ObrT * wT) + (pa.ObrB * wB) + (pa.ObrL
 * wL) + (pa.ObrR * wR);
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (SumProbObr != 0)
                {
                    pa.probT = (pa.ObrT * wT) / SumProbObr;
                    pa.probB = (pa.ObrB * wB) / SumProbObr;
                    pa.probL = (pa.ObrL * wL) / SumProbObr;
                    pa.probR = (pa.ObrR * wR) / SumProbObr;

                }
                else
                {
                    pa.probT = 0;
                    pa.probB = 0;
                    pa.probL = 0;
                    pa.probR = 0;
                }

                //Asign prop to comutlative Function
                pa.inittialCumulativeT();
                pa.inittialCumulativeB();
                pa.inittialCumulativeL();
                pa.inittialCumulativeR();
                randomOUT = pa.getRandom();
                pa.setFootPrint(true);

            //Show Porbability
                this.AddPtextBoxItemT(pa.probT.ToString());
                this.AddPtextBoxItemB(pa.probB.ToString());
                this.AddPtextBoxItemL(pa.probL.ToString());
                this.AddPtextBoxItemR(pa.probR.ToString());

            //check range of probability & choose direction
            if (randomOUT >= 0 && randomOUT < pa.getCDFT())
            {
                pa.stepAction = "UP";
            }
            else if (randomOUT >= pa.getCDFT() && randomOUT <
pa.getCDFB())
            {
                pa.stepAction = "DOWN";
            }
            else if (randomOUT >= pa.getCDFB() && randomOUT <
pa.getCDFL())
            {
                pa.stepAction = "LEFT";
            }
            else if (randomOUT >= pa.getCDFL() && randomOUT <
pa.getCDFR())
            {
                pa.stepAction = "RIGHT";
            }

            //Stamp
            if (this.currentSlot != null && beStartRun == true &&
pa.index != indexFirstPoint)
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            CountStep = CountStep + 1;

                // speed mode (no result)
            if (isSpeedMode)
            {
                // Do ..
                //pa.BackColor = Color.Chocolate;
            }
            else
            {
                // common mode (show result)
                pa.BackColor = Color.Chocolate;
                Deletecolor = true;
            }
        }
        else if (currentSlot == null && beStartRun == false)
        {
            beStartRun = true;
        }
        else if (currentSlot.index == finalIndexPoint)
        {
            beStartRun = false;
            pa.BackColor = Color.Gold;
        }
        else
        {
            CountStep = CountStep + 1;
        }
        this.AddPtextBoxItemS(CountStep.ToString());
        currentSlot = pa;

    }

    private Point findSourcePanel()
    {
        Point pxy = new Point(0, 0);
        bool isbreak = false;
        for (int j = 0; j < 8; j++ )
        {
            if (!isbreak)
            {
                for (int k = 0; k < 8; k++)
                {
                    if (panel_output[k, j].panel_type == 1)
                    {
                        pxy = panel_output[k, j].getPosition();
                        isbreak = true;
                        break;
                    }
                }
            }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                else
                {
                    break;
                }
            }
            return pxy;
        }

        private void assignWeightDirection(String direction ,int w)
        {

            foreach (SubPanel obr in this.panel_output)
            {

               obr.predicDirection = direction;
               obr.weight = w;
               if(direction.Equals("top")){
                   obr.scoreT = w;
                   countT = 1;
               }else if(direction.Equals("buttom")){
                   obr.scoreB = w;
                   countB = 1;
               }
               else if (direction.Equals("left"))
               {
                   obr.scoreL = w;
                   countL = 1;
               }else {
                   obr.scoreR = w;
                   countR = 1;
               }
            }
        }

        private void wTComboBox_SelectedIndexChanged(object sender,
EventArgs e)
        {
            wTChange = 1;
            wTComboBox.Enabled = false;
            gofT = true;
            if (gofT == true && gofB == true && gofL == true && gofR ==
true)
            {
                radioButton1.Enabled = true;
                radioButton2.Enabled = true;
            }
        }

        private void wBComboBox_SelectedIndexChanged(object sender,
EventArgs e)
        {
            wBChange = 1;
            wBComboBox.Enabled = false;
            gofB = true;
            if (gofT == true && gofB == true && gofL == true && gofR ==
true)
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                radioButton1.Enabled = true;
                radioButton2.Enabled = true;
            }
        }

        private void wLComboBox_SelectedIndexChanged(object sender,
EventArgs e)
        {
            wLChange = 1;
            wLComboBox.Enabled = false;
            gofL = true;
            if (gofT == true && gofB == true && gofL == true && gofR ==
true)
            {
                radioButton1.Enabled = true;
                radioButton2.Enabled = true;
            }
        }

        private void wRComboBox_SelectedIndexChanged(object sender,
EventArgs e)
        {
            wRChange = 1;
            wRComboBox.Enabled = false;
            gofR = true;
            if (gofT == true && gofB == true && gofL == true && gofR ==
true)
            {
                radioButton1.Enabled = true;
                radioButton2.Enabled = true;
            }
        }

////////// Used by Delegation //////////
        private void AddPtextBoxItemT(object itemT)
        {
            if (this.pTopValue.InvokeRequired)
                this.pTopValue.Invoke(new
UpdateTextBox(this.AddPtextBoxItemT), itemT);
            else
                this.pTopValue.Text = itemT.ToString();
        }

        private void AddPtextBoxItemB(object itemB)
        {
            if (this.pBottmValue.InvokeRequired)
                this.pBottmValue.Invoke(new
UpdateTextBox(this.AddPtextBoxItemB), itemB);
            else
                this.pBottmValue.Text = itemB.ToString();
        }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        private void AddPtextBoxItemL(object itemL)
        {
            if (this.pLeftValue.InvokeRequired)
                this.pLeftValue.Invoke(new
UpdateTextBox(this.AddPtextBoxItemL), itemL);
            else
                this.pLeftValue.Text = itemL.ToString();
        }

        private void AddPtextBoxItemR(object itemR)
        {
            if (this.pRightValue.InvokeRequired)
                this.pRightValue.Invoke(new
UpdateTextBox(this.AddPtextBoxItemR), itemR);
            else
                this.pRightValue.Text = itemR.ToString();
        }
        private void AddPtextBoxItemS(object itemS)
        {
            if (this.ShowStep.InvokeRequired)
                this.ShowStep.Invoke(new
UpdateTextBox(this.AddPtextBoxItemS), itemS);
            else
                this.ShowStep.Text = itemS.ToString();
        }

        //for show result
        private void AddPtextBoxItemRS(object itemRS)
        {
            if (this.Result.InvokeRequired)
                this.Result.Invoke(new
UpdateTextBox(this.AddPtextBoxItemRS), itemRS);
            else
                this.Result.Text = itemRS.ToString();
        }

        //for show mean
        private void AddPtextBoxItemM(object itemM)
        {
            if (this.ValueofMean.InvokeRequired)
                this.ValueofMean.Invoke(new
UpdateTextBox(this.AddPtextBoxItemM), itemM);
            else
                this.ValueofMean.Text = itemM.ToString();
        }

        //for show Std
        private void AddPtextBoxItemStd(object itemStd)
            {
                if (this.ValueofStd.InvokeRequired)
                    this.ValueofStd.Invoke(new
UpdateTextBox(this.AddPtextBoxItemStd), itemStd);
                else
                    this.ValueofStd.Text = itemStd.ToString();
            }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        private void radioButton1_CheckedChanged(object sender,
 EventArgs e)
        {
            radioButton2.Enabled = false;
            button5.Enabled = true;
            ForStart = 1;
            ForShow = 1;
            isSpeedMode = false;
        }

        private void radioButton2_CheckedChanged(object sender,
 EventArgs e)
        {
            radioButton1.Enabled = false;
            textBox3.Enabled     = true;
            label8.Enabled       = true;
            button7.Enabled      = true;
            ForStart = 1;
            ForNotShow = 1;
            isSpeedMode = true;

        }

        private void button7_Click(object sender, EventArgs e)
        {
            //Input is number,isn't it?
            if (ForNotShow == 1)
            {
                String input2 = this.textBox3.Text;
                try
                {
                    roundCtl = Convert.ToInt32(input2);
                    if (roundCtl >= 0)
                    {
                        countSet = new int[roundCtl];
                        sec = 10;

                            lunch();
                    }
                    else
                    {
                        MessageBox.Show("Input valid integer more than
zero");
                    }
                }
                catch
                {
                    MessageBox.Show("Input integer only");
                }
            }
            button6.Enabled = true;
        }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        private void button6_Click_1(object sender, EventArgs e)
        {
            Result.Clear();
            textBox3.Clear();
            p = 0;
            ShowStep.Clear();
            pTopValue.Clear();
            pBottmValue.Clear();
            pLeftValue.Clear();
            pRightValue.Clear();
            ValueofMean.Clear();
            ValueofStd.Clear();
        }
    }
```

3. คลาสสำหรับสร้างตารางสี่เหลี่ยม

ในคลาสสำหรับสร้างตารางสีเหลี่ยม จะเป็นคลาสที่เก็บฟังก์ชันที่จะถูกเรียกใช้ในฟอร์มที่ 2 เพื่อทำการสร้างตารางสีเหลี่ยมขนาด 8x8 และเพื่อสร้างสิ่งกีดขวาง โดยมีโค้ดดังนี้

```
public class MyPanel
    {
        public SubPanel[,] obcol;
        public SubPanel[,] obrow;
        public SubPanel background;
        public SubPanel[,] block;
        public bool footPrint = false;
        public MyPanel(int i_col ,int j_col ,int i_row, int j_row, int
i_block, int j_block)
        {
            obcol = new SubPanel[i_col, j_col];
            obrow = new SubPanel[i_row, j_row];
            block = new SubPanel[i_block, j_block];
            background = new SubPanel();
        }

        public SubPanel[,] getPanelCol()
        {
            return obcol;
        }

        public SubPanel[,] getPanelRow()
        {
            return obrow;
        }

        public SubPanel[,] getBlockPanel()
        {
            return block;
        }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        public SubPanel getBackGround()
        {
            return background;
        }

        public bool getFootPrintStatus()
        {
            return footPrint;
        }

        public void setFootPrintStatus(bool b)
        {
            footPrint = b;
        }
    }
```

4. คลาสสำหรับเก็บฟังก์ชัน

ในคลาสสำหรับเก็บฟังก์ชัน จะเป็นคลาสที่เก็บฟังก์ชันต่างๆที่จะถูกเรียกใช้ในฟอร์มที่ 2 เพื่อทำการประมวลผลและแสดงผลจากการประมวลผล ซึ่งเป็นคลาสที่ถือได้ว่าเป็นคลาสลูกของคลาสสำหรับสร้างตารางสี่เหลี่ยม โดยมีโค้ดดังนี้

```
public class SubPanel:Panel
    {
        private bool existFootPrint = false;
        private Point point = new Point(0,0);
        public int index = 0;
        public bool isRowOpstruct = false;
        public bool isColOpstruct = false;
        public float probT = 1;
        public float probB = 1;
        public float probL = 1;
        public float probR = 1;
        public int scoreT = 1;
        public int scoreB = 1;
        public int scoreL = 1;
        public int scoreR = 1;
        public int ObrT = 1;
        private double cdfT = 0.0;
        private double cdfB = 0.0;
        private double cdfL = 0.0;
        private double cdfR = 0.0;
        private int weightRemainder = 4;
        public bool isTarget = false;
        public bool isStart = false;
        public int init_dx = 0;
        public int init_dy = 0;
        public int panel_type = 0;
        public int weight = 1;
        public String[] validDirection = {"top","buttom","left",
"right","non"};
        public String predicDirection = "non";
        public String stepAction = "";
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public void setFootPrint(bool print)
{
    existFootPrint = print;
}

public bool getFootPrint()
{
    return existFootPrint;
}

//search FootPrint by index
public void setPosition(int x, int y)
{
    point.X = x;
    point.Y = y;
}

//use for calculate prop
public void decreaseWeightRemainder()
{
    if(weightRemainder >=1 && weightRemainder <=4){
        weightRemainder = weightRemainder - 1;
    }
}

//use for calculate prop
public void increaseWeightRemainder()
{
    if (weightRemainder >=0 && weightRemainder <4){
        weightRemainder = weightRemainder + 1;
    }
}

//use for calculate prop
public void resetWeightRemainder()
{
    weightRemainder = 4;
}

//use for calculate prop
public int getWeightRemainder()
{
    return weightRemainder;
}

//Cumulative
public void inittialCumulativeT()
{
    cdfT = probT;
    cdfT = Convert.ToDouble(String.Format("{0:f2}", cdfT));
}
public void inittialCumulativeB()
{
    cdfB = cdfT + probB;
    cdfB = Convert.ToDouble(String.Format("{0:f2}", cdfB));
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    public void inittialCumulativeL()
    {
        cdfL = cdfB + probL;
        cdfL = Convert.ToDouble(String.Format("{0:f2}", cdfL));
    }
    public void inittialCumulativeR()
    {
        cdfR = cdfL + probR;
        cdfR = Convert.ToDouble(String.Format("{0:f2}", cdfR));
    }
    public double getCDFT()
    {
        return cdfT;
    }
    public double getCDFB()
    {
        return cdfB;
    }
    public double getCDFL()
    {
        return cdfL;
    }
    public double getCDFR()
    {
        return cdfR;
    }

    //get random 0.0-1.0
    public double getRandom()
    {
        Random random = new Random();
        double outP2 = random.NextDouble();
        outP2 = Convert.ToDouble(String.Format("{0:f2}", outP2));
        return outP2;
    }

    public Point getPosition()
    {
        return point;
    }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้