สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบสัญญาณไฟจราจร
TRAFFIC CONTROL SYSTEM

นายธารวิทย์ ดิษยวงศ์
นายสัญชัย รุ่งเรืองชูสกุล

เลขหมู่.....
เลขทะเบียน 42820
วัน, เดือน, ปี 10 มิ.ย. 2545

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า, ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสัญญาณไฟจราจร

TRAFFIC CONTROL SYSTEM

โดย

นายธารวิทย์ ดิษยวงศ์

นายสัญชัย รุ่งเรืองชูสกุล

อาจารย์ที่ปรึกษา

อ.สมเกียรติ วังศิริพิทักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสัญญาณไฟจราจร

TRAFFIC-CONTROL SYSTEM

ผู้จัดทำ

1. นาย ธารวิทย์ ดิษยวงศ์ รหัสประจำตัว 40010318
2. นาย สัญชัย รุ่งเรืองชูสกุล รหัสประจำตัว 40010836

_____ อาจารย์ที่ปรึกษา

(อ. สมเกียรติ วังศิริพิทักษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสัญญาณไฟจราจร

นายธารวิทย์ ดิษยวงศ์ 40010318
นายสัญชัย รุ่งเรืองชูสกุล 40010836
อ.สมเกียรติ วังศิริพิทักษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ทุกวันนี้ การคมนาคมเป็นสิ่งที่จำเป็นมากสำหรับประชาชนทั่วไป โดยอาจจะถือว่าเป็นสาธารณูปโภคอย่างหนึ่งเลยก็ได้ แต่ในเมืองใหญ่ๆ แล้วการจราจรถือได้ว่าเป็นหัวใจของการเดินทางเลยก็ว่าได้ ดังนั้นถ้ามีการจัดการด้านการจราจรอย่างมีประสิทธิภาพก็จะทำให้การเดินทางมีประสิทธิภาพมากยิ่งขึ้นด้วย แต่ปัญหาที่เกิดขึ้นในการจัดการด้านการจราจรนั้นมีหลายประการ และการแก้ไขใช้เวลานานในบางทีอาจจะไม่พอกับการเปลี่ยนแปลงในอนาคต หรือการแก้ปัญหานั้นอาจจะไม่ถูกต้องเลยก็ได้

ในประเทศไทยเองการจัดการด้านการจราจรยังไม่ดีมากนัก ดังจะเห็นได้จากการติดขัดของรถเป็นเวลานาน ดังนั้นจึงจำเป็นจะต้องมีการจัดการด้านการจราจรอย่างเหมาะสม ซึ่งการจัดการด้านการจราจรนั้นสามารถแบ่งออกเป็น 2 ประเภทคือ

1. ในระยะสั้น คือการควบคุมสัญญาณไฟจราจร ให้ปล่อยรถได้อย่างมีประสิทธิภาพ
2. ในระยะยาว คือการตัดถนนเพิ่ม เป็นการกระจายปริมาณรถไปสู่ถนนอื่นๆ

ในโครงงานที่จัดทำขึ้นเป็นโปรแกรมที่ใช้สำหรับจำลองการทำงานของระบบสัญญาณไฟจราจร โดยโปรแกรมจะแบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนของ Editor จะเป็นส่วนของการป้อนข้อมูลของโปรแกรม และในส่วนของการคำนวณไฟสัญญาณไฟจราจร โดยการคำนวณสัญญาณไฟจราจรจะใช้ AI (Artificial Intelligence) ในการคำนวณ โดยจะแสดงผลลัพธ์ที่ได้เป็นสีของถนน การนำ AI มาช่วยในการคำนวณจะช่วยให้สามารถแก้ไขปัญหาทางการจราจรได้ดียิ่งขึ้น

ลักษณะโดยทั่วไปของโปรแกรมจะเป็นโปรแกรมที่มีลักษณะ User friendly โดยจะใช้ Tools ของ Java Development Kit Version 1.3 ในการพัฒนาโปรแกรม โปรแกรมดังกล่าวเป็นโปรแกรมที่มีลักษณะของการใช้ OOP (Object Oriented Programming) และจะทำการเก็บไฟล์ในรูปของ Vector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Traffic Control System

Tharawit Ditsayawong

Sunchai Rungruengchoosakul

Mr. Somkiat Wangsiripitak Advisor

ABSTRACT

Nowadays, transportation is very important for all people. It is a kind of significant public utility which the government should provide for their citizen, especially in big cities. By all mean of transportation in BKK, car seems to be the most important vehicle. Therefore, the more effort put into traffic control the better is the transportation. The traffic control could be done by many ways, both in short term and long term.

Thailand's traffic control is not so good as you can see the traffic jam on many road in BKK. The problem needs to be solved by using these criteria

1. Short term problem solving such as control traffic light in order keep the traffic runs smoothly.
2. Long term problem solving such as building more roads to support the increasing in amount of cars.

This project prepares the traffic simulation program. It is divided in to 2 parts. Part I Editor, It is used for building the road in cities. Part II Traffic Light Control System AI is used to calculate the frequency and pattern of traffic light.

The result will be shown in computer graphics. By many experiment, it is found that AI is very useful in controlling the traffic light.

JAVA Development Kit Version 1.3 develops this project's program, which is very easy to use. The program written by using OOP and the files is kept in the vector form.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ สมเกียรติ วังศิริพิทักษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ ให้คำปรึกษา ชี้แนะแนวทางการทำงาน และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณ สำนักงานคณะกรรมการจัดระบบจราจรทางบก สำนักนายกรัฐมนตรี สำหรับข้อมูลที่เป็นประโยชน์ รวมทั้งแนวทางในการพัฒนาโครงงาน

ขอขอบคุณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และสถานศึกษาในอดีต สำหรับโอกาสดีๆ ทางการศึกษาที่ทำให้ผู้เขียนได้มีโอกาสถึงทุกวันนี้

ขอขอบคุณผู้ที่ได้มีส่วนร่วมในการทำงาน และช่วยเหลือทั้งแรงกาย แรงใจ และความคิด ที่ช่วยให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

และที่สำคัญที่สุดต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา และผู้มีพระคุณ อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ กำลังใจที่ดี และคอยเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

ธารวิทย์ ดิษยวงศ์
สัญชัย รุ่งเรืองชูสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญรูปภาพ IX
สารบัญตาราง XII
บทที่ 1 บทนำ 1
1.1 ความสำคัญของโครงงาน 1
1.2 จุดประสงค์ของโครงงาน 2
1.3 ขอบเขตของโครงงาน 2
1.4 ประโยชน์ของโครงงาน 3
1.5 ความน่าเชื่อถือของโครงงาน 3
1.6 วิธีการดำเนินงาน 4
บทที่ 2 ความรู้พื้นฐานของโปรแกรม 5
2.1 ทฤษฎีทาง Artificial Intelligent 5
2.2 Object Oriented Design & Object Oriented Programming 5
2.3 Data Structure 6
2.3.1 Link List 6
2.3.2 Queue 6
2.4 Traffic Engineering 7
2.4.1 ปัจจัยที่ผลต่อการจราจร 7
2.4.2 นิยามของรถติด 8
2.4.3 ความเร็วเฉลี่ยของถนน 9
2.4.4 การปล่อยรถที่ทางแยก 11
2.4.5 ลักษณะการวิ่งของรถ 12
2.4.6 ลักษณะโดยทั่วไปของถนน 13
2.4.7 ลักษณะโดยทั่วไปของทางแยก 15
2.4.8 สัญญาณไฟจราจร 15
บทที่ 3 การออกแบบ 16
3.1 ข้อกำหนดที่ใช้ในการออกแบบ (การสมมติเหตุการณ์ที่เกิดขึ้นในระบบ) 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

3.2 การออกแบบโปรแกรม 16

3.2.1 Use Case Diagram 16

3.2.2 การออกแบบคลาส (Class Design) 17

3.2.3 Data Dictionary 19

3.2.3.1 คลาส Controller 20

3.2.3.2 คลาส CrossRoad (คลาสทางแยก) 25

3.2.3.3 คลาสสามแยก 32

3.2.3.4 คลาสสี่แยก 34

3.2.3.5 คลาสห้าแยก 35

3.2.3.6 คลาสถนน 37

3.2.3.7 คลาสถนนทางเดียว 44

3.2.3.8 คลาสถนนสองทาง 46

3.2.3.9 คลาสสัญญาณไฟจราจร 52

3.2.3.10 คลาส Index 54

3.2.3.11 คลาส OpenFile 56

3.2.3.12 คลาส SaveFile 56

3.2.3.13 คลาส Cal 58

3.2.3.14 คลาส RunThread 63

3.2.3.15 คลาส RunNormalThread 68

3.2.3.16 คลาส RunBestSelectThread 68

3.2.3.17 คลาส RunBestTimeThread 70

3.2.3.18 คลาส RunSelectRoadThread 70

3.2.4 Sequential Diagram 72

3.2.4.1 Sequential Diagram ของการสร้างและแก้ไข 72

3.2.4.2 Sequential Diagram ของการบันทึกไฟล์ 74

3.2.4.3 Sequential Diagram ของการเปิดไฟล์ 76

3.2.4.4 Sequential Diagram ของการ Run โปรแกรม 77

3.2.5 การออกแบบหน้าจอ (Screen Design) 80

3.2.6 การออกแบบ AI ที่ใช้ในการคำนวณสัญญาณไฟจราจร 81

3.2.6.1 การคำนวณสัญญาณไฟจราจรแบบปล่อยปริมาณรถได้มากที่สุด 82

3.2.6.2 การคำนวณสัญญาณไฟแบบปล่อยสัญญาณไฟจราจรให้คุ้มค่าที่สุด 84

3.2.6.3 การคำนวณสัญญาณไฟจราจรแบบเลือกถนนที่ต้องการให้ติดน้อยที่สุด 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

บทที่ 4 การเขียนโปรแกรม 88

4.1 อธิบายคลาสต่างๆ ที่ใช้ในโปรแกรม 88

4.1.1 Package kernel 90

4.1.1.1 คลาส Controller 90

4.1.1.2 คลาส Road 90

4.1.1.3 คลาส Road1 90

4.1.1.4 คลาส Road2 90

4.1.1.5 คลาส CrossRoad 90

4.1.1.6 คลาส Cross3 90

4.1.1.7 คลาส Cross4 91

4.1.1.8 คลาส Cross5 91

4.1.1.9 คลาส MessageBox 91

4.1.1.10 คลาส Index 91

4.1.1.11 คลาส TrafficLight 91

4.1.2 Package display 91

4.1.2.1 คลาส Cal 92

4.1.3 Package io 92

4.1.3.1 คลาส SaveFile 92

4.1.3.2 คลาส OpenFile 92

4.1.4 Package Run 92

4.1.4.1 คลาส RunThread 92

4.1.4.2 คลาส RunNormalThread 92

4.1.4.3 คลาส RunBestSelectThread 92

4.1.4.4 คลาส RunBestTimeThread 93

4.1.4.5 คลาส RunSelectRoadThread 93

4.1.4.6 Sub Package ui 93

4.1.4.6.1 คลาส CrossDialog 93

4.1.4.6.2 คลาส RoadDialog 94

4.1.4.6.3 คลาส LowPercentDialog 94

4.1.5 Package monitoring 95

4.1.5.1 คลาส Monitoring 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

4.1.6 Package ui 96
4.1.6.1 คลาส AboutDialog 96
4.1.6.2 คลาส BaseDialog 96
4.1.6.3 คลาส crossConnectDialog 96
4.1.6.4 คลาส EditorFrame 97
4.1.6.5 คลาส HelpDialog 98
4.1.6.6 คลาส Interface 98
4.1.6.7 คลาส MapFrame 99
4.1.6.8 คลาส MouseMotionListen 100
4.1.6.9 คลาส ObjectDialog 100
4.1.6.10 คลาส PropertiesCrossDialog 100
4.1.6.11 คลาส PropertiesProgramDialog 101
4.1.6.12 คลาส PropertiesProjectDialog 102
4.1.6.13 คลาส PropertiesRoadDialog 103
4.1.6.14 คลาส RunAIDialog 104
4.1.6.15 คลาส RunningFrame 105
4.1.6.16 คลาส RunNormalDialog 106
4.1.6.17 คลาส ShowRoad 106
4.1.6.18 คลาส ShowTrafficStep 107
4.1.6.19 คลาส SplashScreen 107
4.1.6.20 Sub package io 108
4.1.6.20.1 คลาส SaveOutput 108
4.2 ปัญหาและการแก้ไข 108
บทที่ 5 การทดสอบโครงงาน 112
5.1 ผลลัพธ์ที่ได้จากการทดลอง 112
5.2 ผลการทดลองที่ได้ 113
5.2.1 ผลการจำลองการทำงานของสัญญาณไฟจราจรแบบปรกติ (ตั้งค่าตามเวลา) 123
5.2.2 ผลการทดลองระบบสัญญาณไฟจราจรแบบเลือก
จังหวะสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด (Best Select) 127
5.2.3 ผลการทำงานของระบบสัญญาณไฟจราจรแบบเลือก
คำนวณสัญญาณไฟจราจรที่ดีที่สุด (Best Time) 131

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

5.2.4 ผลการทำงานของระบบสัญญาณไฟจราจรแบบเลือก
ถนนที่ต้องการปล่อยรถได้มากที่สุด (Select Road) 135
5.2.5 สรุปผลการทำงานโดยรวมของการคำนวณสัญญาณไฟจราจร 141
บทที่ 6 สรุป 144
6.1 สรุปโครงงาน 144
6.1.1 สรุปการทำงานในส่วนของโมเดลที่ใช้ในโปรแกรม 144
6.1.2 สรุปการทำงานในส่วนของ Editor 144
6.1.3 สรุปการทำงานในส่วนของการจำลองการทำงานของสัญญาณไฟ 144
6.1.4 สรุปการทำงานโดยรวมของระบบ 146
6.2 แนวทางการพัฒนาต่อ 146
ภาคผนวก ก. ภาษา JAVA 148
ก.1 ขั้นตอนการติดตั้ง 149
ก.2 สาเหตุการใช้ภาษา JAVA 149
ภาคผนวก ข. ระบบการจราจร (Traffic System) 151
ข.1 การพัฒนาของทางแยก (Cross Road Generation) 151
ข.2 Area Traffic Control (ATC) 153
ข.3 สภาพระบบการจราจรของกรุงเทพมหานครในปัจจุบัน 156
ข.4 อนาคตของการจราจรในกรุงเทพมหานคร 157
บรรณานุกรม 159

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้าที่

รูปที่ 1-1 แสดงลักษณะของทางแยกที่มีการคำนวณระบบสัญญาณไฟ 2
รูปที่ 2-1 แสดงลักษณะของข้อมูลแบบ Link List 6
รูปที่ 2-2 แสดงลักษณะของข้อมูลแบบ Queue 7
รูปที่ 2-3 แสดงนิยามของรถติดตามที่ได้กำหนดไว้ 8
รูปที่ 2-4 แสดงลักษณะของถนนที่มีความเร็วไม่เท่ากันทั้งเส้น 10
รูปที่ 2-5 แสดงการหาค่าของ CarHeight 11
รูปที่ 2-6 แสดงค่าที่เกี่ยวข้องในการเคลื่อนย้ายรถจากถนน A ไปยังถนน B 12
รูปที่ 2-7 แสดงลักษณะการจำลองการวิ่งของรถในแบบที่ง่ายที่สุด 13
รูปที่ 2-8 แสดงลักษณะโมเดลของถนน 14
รูปที่ 2-9 แสดงโมเดลของสามแยก 15
รูปที่ 3-1 แสดง Use Case Diagram 17
รูปที่ 3-2 แสดงภาพรวมของการออกแบบคลาส 18
รูปที่ 3-3 แสดงลักษณะของคลาสที่ได้ทำการออกแบบไว้ในมาตรฐานของ UML 19
รูปที่ 3-4 แสดงการกำหนดการแสดงผลของถนนว่าเป็นเช่นใด 23
รูปที่ 3-5 แสดงลักษณะของ op และ time ที่ใช้ในทางแยก 27
รูปที่ 3-6 แสดงลักษณะของการเชื่อมต่อกันของทางแยก และถนนเข้าด้วยกัน 31
รูปที่ 3-7 แสดงการเชื่อมต่อของถนนกับถนน 31
รูปที่ 3-8 แสดงการโมเดลสามแยกในส่วนของการแสดงผลของข้อมูล 33
รูปที่ 3-9 แสดง index ของถนนในสามแยก 33
รูปที่ 3-10 แสดง index ของถนนในสี่แยก 34
รูปที่ 3-11 แสดงลักษณะโมเดลของสี่แยกในการแสดงผลที่องศาเป็น 0 35
รูปที่ 3-12 แสดงรูปของการแสดงผลของห้าแยกที่องศาเป็น 0 36
รูปที่ 3-13 แสดง index ของถนนในห้าแยก 37
รูปที่ 3-14 แสดงลักษณะของถนนที่มี Queue และไม่มี Queue 40
รูปที่ 3-15 แสดงลักษณะของการแสดงผลของถนนทางเดียว 45
รูปที่ 3-16 แสดงโมเดลของถนนทางเดียวในส่วนของ Data Structure 46
รูปที่ 3-17 แสดงโมเดลของถนนทางเดียวในส่วนของ Method 46
รูปที่ 3-18 แสดงลักษณะของการแสดงผลของถนนสองทาง 49
รูปที่ 3-19 แสดงลักษณะ Data Structure ของถนนสองทาง 51
รูปที่ 3-20 แสดงลักษณะการทำงานของ Method ในถนนสองทาง 52
รูปที่ 3-21 แสดงการแสดงผลของวัตถุสัญญาณไฟจราจร 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ 3-22 แสดงลักษณะของการเปิดไฟล์ และการบันทึกไฟล์อย่างง่ายๆ 58
รูปที่ 3-23 แสดงความสัมพันธ์ของแกนการทำงานทั้งสองแกน 60
รูปที่ 3-24 แสดงลักษณะของการแปลงค่าจากแกน World ไปยังแกน Screen 61
รูปที่ 3-25 แสดงขั้นตอนการแปลงค่าจากแกนของแผนที่ไปเป็นแกนของหน้าจอ 61
รูปที่ 3-26 แสดงลักษณะของการหมุนจุดที่แกน World 62
รูปที่ 3-27 แสดงขั้นตอนการหาสมการของการหมุนที่จุดใดๆ ในแผนที่ 62
รูปที่ 3-28 แสดง Flow Chart ใน Method run() ของคลาส RunThread 67
รูปที่ 3-29 แสดง Flow Chart การคำนวณสัญญาณไฟจราจร 68
รูปที่ 3-30 แสดง Sequential Diagram ของการสร้างและแก้ไขแผนที่ 73
รูปที่ 3-31 แสดง Sequential Diagram ของการบันทึกไฟล์ 75
รูปที่ 3-32 แสดงลักษณะของ Sequential Diagram ของการเปิดไฟล์ 76
รูปที่ 3-33 แสดงลักษณะของ Sequential Diagram ของการจำลองการทำงานของสัญญาณไฟจราจร 78
รูปที่ 3-34 แสดงหน้าจอคร่าวๆ ของโปรแกรม 80
รูปที่ 3-35 แสดงลักษณะของหน้าจอการจำลองการทำงานของโปรแกรม 81
รูปที่ 3-36 แสดงการทำงานอย่างคร่าวๆ ของการคำนวณสัญญาณไฟ
แบบเลือกสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด 82
รูปที่ 3-37 แสดงลักษณะของการคำนวณสัญญาณไฟจราจร
แบบเลือกสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด 83
รูปที่ 3-38 แสดงลักษณะของการคำนวณสัญญาณไฟจราจร
แบบหาเวลาที่ดีที่สุดในการปล่อยสัญญาณไฟจราจร 85
รูปที่ 3-39 แสดงลักษณะของการคำนวณสัญญาณไฟจราจรแบบเลือกถนนที่ต้องการให้รถติดน้อยที่สุด 86
รูปที่ 4-1 แสดงลักษณะของคลาส MessageBox ในการข้อความ 91
รูปที่ 4-2 แสดงลักษณะของหน้าจอของคลาส CrossDialog 94
รูปที่ 4-3 แสดงลักษณะของหน้าจอของ RoadDialog 94
รูปที่ 4-4 แสดงลักษณะหน้าจอของ LowPercentDialog 95
รูปที่ 4-5 แสดงหน้าจอของคลาส AboutDialog 96
รูปที่ 4-6 แสดงหน้าจอของ BaseDialog 96
รูปที่ 4-7 แสดงหน้าจอของคลาส crossConnectDialog 97
รูปที่ 4-8 แสดงลักษณะหน้าจอของ EditorFrame 97
รูปที่ 4-9 แสดงลักษณะหน้าจอของ HelpDialog 98
รูปที่ 4-10 แสดงลักษณะหน้าจอของ Interface 99
รูปที่ 4-11 แสดงหน้าจอของ MapFrame 99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ 4-12 แสดงหน้าจอของ ObjectDialog 100
รูปที่ 4-13 แสดงหน้าจอของ CrossDialog 101
รูปที่ 4-14 แสดงลักษณะหน้าจอของ PropertiesProgramDialog 102
รูปที่ 4-15 แสดงลักษณะหน้าจอของ PropertiesProjectDialog 103
รูปที่ 4-16 แสดงลักษณะของหน้าของ PropertiesRoadDialog 104
รูปที่ 4-17 แสดงหน้าจอของ RunAIDialog 104
รูปที่ 4-18 แสดงหน้าจอของ RunningFrame 105
รูปที่ 4-19 แสดงหน้าจอของ RunNormalDialog 106
รูปที่ 4-20 แสดงหน้าจอของ ShowRoad 106
รูปที่ 4-21 แสดงหน้าจอของ ShowTrafficStep 107
รูปที่ 4-22 แสดงหน้าจอของ SplashScreen 107
รูปที่ 5-1 แสดงลักษณะของหน้าจอจาก MapFrame 112
รูปที่ 5-2 แสดงลักษณะของแผนที่ที่ 1 ที่ใช้ในการทดสอบ 113
รูปที่ 5-3 แสดงรายละเอียดของสัญลักษณ์ที่ใช้ในรายละเอียดของทางแยก 114
รูปที่ 5-4 แสดงลักษณะของแผนที่ที่ 2 115
รูปที่ 5-5 แสดงลักษณะของแผนที่ที่ 3 118
รูปที่ ก-1 แสดงลักษณะโดยรวมของภาษา JAVA 148
รูปที่ ข-1 แสดงลำดับขั้นของการพัฒนาของทางแยก 152
รูปที่ ข-2 แสดงแผนที่ของทางแยกที่มีการติดตั้งระบบ ATC ของกรุงเทพมหานคร 156
รูปที่ ข-3 แสดงระยะของระบบ ATC ที่ใช้ในกรุงเทพมหานคร 158

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 3-1 แสดงรายละเอียดของ Attribute ของคลาส Controller 22
ตารางที่ 3-2 แสดงรายละเอียดของ Method ของคลาส Controller 25
ตารางที่ 3-3 แสดงรายละเอียดของ Attribute ของคลาสทางแยก 27
ตารางที่ 3-4 แสดงรายละเอียดของ Method ของคลาสทางแยก 30
ตารางที่ 3-5 แสดงรายละเอียดของ Attribute ของคลาสสามแยก 32
ตารางที่ 3-6 แสดงรายละเอียดของ Attribute ของคลาสสี่แยก 34
ตารางที่ 3-7 แสดงรายละเอียดของ Attribute ของคลาสห้าแยก 35
ตารางที่ 3-8 แสดงรายละเอียด Attribute ของคลาสถนน 39
ตารางที่ 3-9 แสดงรายละเอียดของ Method ของคลาสถนน 44
ตารางที่ 3-10 แสดงรายละเอียดของ Attribute ของคลาสถนนทางเดียว 45
ตารางที่ 3-11 แสดงรายละเอียดของ Attribute ของคลาสถนนสองทาง 48
ตารางที่ 3-12 แสดงรายละเอียดของ Method ของคลาสถนนสองทาง 51
ตารางที่ 3-13 แสดงรายละเอียดของ Attribute ของคลาสสัญญาณไฟจราจร 53
ตารางที่ 3-14 แสดงรายละเอียดของ Method ของคลาสสัญญาณไฟจราจร 53
ตารางที่ 3-15 แสดงรายละเอียดของ Method ของคลาส Index 55
ตารางที่ 3-16 แสดงรายละเอียดของ Method ของคลาส OpenFile 56
ตารางที่ 3-17 แสดงรายละเอียดของ Method ของคลาส SaveFile ในระบบ 58
ตารางที่ 3-18 แสดงรายละเอียดของ Method ของคลาส Cal 60
ตารางที่ 3-19 แสดงรายละเอียด Attribute ของคลาส RunThread 64
ตารางที่ 3-20 แสดงรายละเอียดของ Method ของคลาส RunThread 66
ตารางที่ 3-21 แสดงรายละเอียดของ Attribute ของคลาส RunBestSelectThread 68
ตารางที่ 3-22 แสดงรายละเอียดของ Method ของคลาส RunBestSelectThread 69
ตารางที่ 3-23 แสดงรายละเอียดของ Attribute ของคลาส RunSelectRoadThread 70
ตารางที่ 3-24 แสดงรายละเอียดของ Method ของคลาส RunSelectRoadThread 71
ตารางที่ 4-1 แสดง Package ต่างๆ ของโปรแกรม 89
ตารางที่ 4-2 แสดงปัญหาและการแก้ไขของระบบ 111
ตารางที่ 5-1 แสดงรายละเอียดของถนนในแผนที่ที่ 1 113
ตารางที่ 5-2 แสดงรายละเอียดของทางแยกในแผนที่ที่ 1 114
ตารางที่ 5-3 แสดงรายละเอียดของถนนในแผนที่ที่ 2 116
ตารางที่ 5-4 แสดงรายละเอียดของทางแยกในแผนที่ที่ 2 117
ตารางที่ 5-5 แสดงรายละเอียดของถนนในแผนที่ที่ 3 119

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

ตารางที่ 5-6 แสดงรายละเอียดของแผนที่ที่ 3 122
ตารางที่ 5-7 แสดงผลการทำงานของสัญญาณไฟจราจรแบบปรกติในแผนที่ที่ 1 124
ตารางที่ 5-8 แสดงผลการทำงานของสัญญาณไฟจราจรแบบปรกติในแผนที่ที่ 2 125
ตารางที่ 5-9 แสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบปรกติในแผนที่ที่ 3 126
ตารางที่ 5-10 แสดงผลการทำงานของสัญญาณไฟจราจรแบบ
เลือกจังหวะสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุดในแผนที่ที่ 1 128
ตารางที่ 5-11 แสดงผลการทำงานของสัญญาณไฟจราจรแบบ
เลือกจังหวะสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุดในแผนที่ที่ 2 129
ตารางที่ 5-12 แสดงผลการทำงานของสัญญาณไฟจราจรแบบ
เลือกจังหวะสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุดในแผนที่ที่ 3 130
ตารางที่ 5-13 แสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบ
เลือกคำนวณเวลาของสัญญาณไฟจราจรที่ดีที่สุดในแผนที่ที่ 1 132
ตารางที่ 5-14 แสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบ
เลือกสัญญาณไฟจราจรที่ดีที่สุดในแผนที่ที่ 2 133
ตารางที่ 5-15 แสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบ
เลือกสัญญาณไฟจราจรที่ดีที่สุดในแผนที่ที่ 3 135
ตารางที่ 5-16 แสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบ
เลือกเส้นทางที่ต้องการปล่อยรถได้มากที่สุด (เลือกถนนปลายเปิด) 137
ตารางที่ 5-17 แสดงผลของรายละเอียดถนนที่เลือก (ถนนปลายเปิด) ในแผนที่ที่ 2
เปรียบเทียบกับสัญญาณไฟจราจรแบบปรกติ 138
ตารางที่ 5-18 แสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบ
เลือกเส้นทางที่รถติดน้อยที่สุดในแผนที่ที่ 2 (เลือกถนนกลางแผนที่) 139
ตารางที่ 5-19 แสดงค่า Car Rate ของถนนที่เลือก (ถนนกลางแผนที่) ในแผนที่ที่ 2
เปรียบเทียบกับสัญญาณไฟจราจรแบบปรกติ 140
ตารางที่ 5-20 แสดงผลการทำงานของอัลกอริทึมโดยรวมที่เปอร์เซ็นต์ของรถเป็น 0.05 141
ตารางที่ 5-21 แสดงผลการทำงานของอัลกอริทึมโดยรวมที่เปอร์เซ็นต์ของรถเป็น 0.5 142
ตารางที่ 5-22 แสดงผลการทำงานของอัลกอริทึมโดยรวมที่เปอร์เซ็นต์ของรถเป็น 5 142
ตารางที่ 6-1 แสดงผลสรุปของอัลกอริทึมที่ได้ออกแบบมาเปรียบเทียบกัน 145
ตารางที่ ข-1 แสดงรายชื่อทางแยกที่มีการติดตั้งระบบ ATC แล้ว 155

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันนี้การคมนาคมเป็นสิ่งที่สำคัญมากในชีวิตประจำวัน อาจจะถือได้ว่าเป็นสาธารณูปโภคชนิดหนึ่งก็ได้ เนื่องจากว่าคนทุกคนต่างมีกิจวัตรประจำวันที่ต้องเกี่ยวข้องกับการคมนาคมเป็นส่วนมาก ในเมืองไทยเองก็มีการคมนาคมได้หลายทาง แต่ว่าการคมนาคมดังกล่าวอาจจะไม่เหมาะสมกับคนส่วนมาก หรือในบางสถานที่ไม่สะดวกในการเดินทางแบบดังกล่าว แต่การคมนาคมที่เป็นพื้นฐานและที่สำคัญที่สุดคือ การใช้ระบบการจราจร แต่เนื่องจากว่าในเมืองไทยประสบปัญหาด้านดังกล่าวเป็นอย่างมาก ทำให้การจราจรเป็นไปอย่างล่าช้า ดังนั้นเราจึงจำเป็นต้องศึกษา และทำการวิเคราะห์ว่าระบบการจราจรที่ล่าช้านั้นมีสาเหตุมาจากอะไรบ้าง ซึ่งเราสามารถสรุปได้ง่ายๆ ว่า ปริมาณรถที่อยู่บนถนนมีปริมาณมากเกินขนาดของถนนที่จะรับได้ (การวิเคราะห์ปัญหาดังกล่าว เราจะวิเคราะห์กันอย่างละเอียดในบทที่ 2) แต่สาเหตุดังกล่าวมีได้หลายทางแก้ แต่สิ่งที่ควรจะทำเป็นอย่างแรกก็คือการลดปริมาณรถลงในถนน โดยหันมารณรงค์ใช้รถโดยสารให้มากขึ้น เป็นต้น แต่ในเมื่อไม่สามารถทำในขั้นตอนแรกได้ เราจึงต้องอาศัยเทคโนโลยีมาช่วยจัดการการจราจรดังกล่าว ให้เหมาะสมยิ่งขึ้น โดยเราจะแบ่งการจัดการออกเป็น 2 ประเภท คือ

1. การจัดการเกี่ยวกับจัดสร้างถนนขึ้นมาใหม่
2. การจัดการเกี่ยวกับระบบสัญญาณไฟจราจร

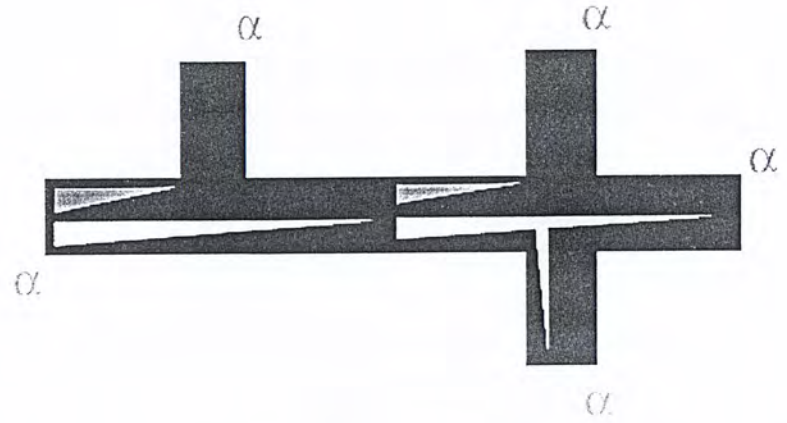
การจัดการทั้ง 2 อย่างดังกล่าวมีข้อแตกต่างกันมาก กล่าวคือการจัดการในข้อ 1. จะต้องเสียงบประมาณอย่างมาก และในบางทีอาจจะแก้ปัญหาในทันทีไม่ได้ เป็นการแก้ปัญหาระยะยาว ต้องเสียเวลาในการจัดการอย่างมาก ส่วนในข้อ 2. นั้นเป็นการจัดการที่ง่ายมากยิ่งขึ้น เสียงบประมาณน้อยกว่า แต่เป็นการแก้ปัญหาในระยะสั้นเท่านั้น เพราะฉะนั้นในการจัดการกับระบบจราจรนั้น จะต้องจัดการทั้ง 2 อย่างควบคู่กันไป เพื่อให้เกิดประสิทธิภาพสูงสุด

1.1 ความสำคัญของโครงงาน

ในการก่อสร้างทางเพิ่มนั้นเป็นปัญหาอย่างมาก เนื่องจากว่าต้องใช้งบประมาณที่มาก และยังใช้ระยะเวลาในการก่อสร้างที่นาน อีกทั้งยังกีดขวางการจราจรเดิมด้วย ทำให้เกิดการจราจรติดขัดที่หนักมากขึ้น เราจึงต้องวางแผนการตัดถนนให้ละเอียดรอบคอบก่อนที่จะลงมือปฏิบัติเพื่อให้เกิดประสิทธิภาพสูงสุดในการทำงาน ถ้าเราสามารถจำลองการทำงานดังกล่าวให้บนเครื่องคอมพิวเตอร์แล้วลองทดสอบดูประสิทธิภาพของระบบก่อนที่จะลงมือปฏิบัติจริงได้ก็จะเกิดประโยชน์อย่างมาก และเป็นการลงทุนที่น้อยมากเมื่อเทียบกับการลงมือปฏิบัติจริง อนึ่งการทำงานบนเครื่องคอมพิวเตอร์ถือว่าเป็นการทำงานที่ง่ายและสะดวกมากในการทำงาน

ในการคำนวณสัญญาณไฟจราจรนั้นเป็นสิ่งที่สำคัญมากดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-1 แสดงลักษณะของทางแยกที่มีการคำนวณระบบสัญญาณไฟ

จากรูปที่ 1-1 เราจะมองเฉพาะระบบที่สนใจว่าเป็นอย่างไร โดยเราจะกำหนดเอาไว้ให้ส่วนขอบ (Boundary) ของระบบสัญญาณไฟเป็น Infinity (∞) ซึ่งหมายความว่ารถจะสามารถเข้าออกจากระบบได้โดยอิสระ และไม่มีการจอดรถค้างไว้ในถนน ซึ่งจะทำให้รถสามารถเคลื่อนที่ได้อย่างคล่องตัว จากการที่สัญญาณไฟมีการคำนวณ เราจะต้องอาศัยเทคนิคของ Artificial Intelligent (AI) มาช่วยในการคำนวณ ซึ่งการคำนวณจะทำให้ระบบมีประสิทธิภาพมากขึ้น โดยคาดหวังว่าจะทำให้ระบบช่วยกระจายปริมาณรถไปยังทุกๆ ถนน ทำให้รถสามารถเคลื่อนที่รถได้อย่างมีประสิทธิภาพมากขึ้น

1.2 จุดประสงค์ของโครงงาน

เพื่อเป็นต้นแบบในการพัฒนาเป็นโปรแกรมที่ช่วยในการจัดระบบการจราจร เช่น การวางแผนในการตัดถนน ทดลองคำนวณสัญญาณไฟจราจรโดยใช้ AI เพื่อให้รถติดน้อยที่สุดโดยจะใช้การพัฒนาโปรแกรมในแบบ Object Oriented Programming (OOP) และโดยใช้ภาษา Java (อาศัย tools ที่มีชื่อว่า Java Development Kit Version 1.2.1 ในการพัฒนาโปรแกรม) โดยโปรแกรมที่พัฒนาขึ้นมาจะมีลักษณะของ User-friendly และจะทำงานแบบ off line มีลักษณะเป็น Application แบบ Stand Alone

1.3 ขอบเขตของโครงงาน

เป็นโปรแกรมที่ใช้จำลองการทำงานของระบบสัญญาณไฟจราจร ซึ่งโปรมแกรมจะแบ่งการทำงานเป็นสองแบบ คือ

1. แบบธรรมดา เป็นเหมือนสัญญาณไฟทั่วไป จะเปิดจังหวะสัญญาณไฟจราจรตามเวลาที่ได้ตั้งเอาไว้
3. แบบใช้ AI จะใช้ AI เข้ามาช่วยคำนวณจังหวะสัญญาณไฟจราจรในการปล่อยรถ โดยจะมีผลทำให้รถติดน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโปรแกรมที่สร้างขึ้นจะแบ่งการพัฒนาออกเป็นสองส่วนคือ

- ส่วน Editor ซึ่งจะเป็นส่วนของการสร้างและแก้ไขข้อมูลลงในโปรแกรม
- ส่วนของ AI ซึ่งจะใช้ในการคำนวณการกำหนดจังหวะสัญญาณไฟจราจร

โดยการพัฒนาโปรแกรมจะใช้การพัฒนาโปรแกรมแบบ OOP เพื่อสามารถนำกลับมาใช้และทำการแก้ไขโปรแกรมได้ง่าย

1.4 ประโยชน์ของโครงงาน

เพื่อนำไปเป็นต้นแบบสำหรับการพัฒนาโปรแกรมให้เป็นโปรแกรมสำหรับช่วยในการจัดระบบการจราจรที่สามารถใช้งานได้จริง โดยโปรแกรมจะแบ่งการทำงานออกเป็น 2 ส่วน คือ

1. ส่วนของ Editor จะมีประโยชน์คือ ช่วยในการวิเคราะห์การตัดถนน และการสร้างถนนว่าจะได้รับประโยชน์สูงสุดในการใช้งานหรือไม่ และควรจะวางแผนในการตัดถนนที่ส่วนใด จึงจะช่วยแก้ไขปัญหาในการจราจรได้เกิดประโยชน์สูงสุด และคุ้มค่าที่สุด

2. ในส่วนของการ Run โปรแกรมจะแบ่งการทำงานออกเป็น 2 ส่วน คือ
   - ส่วนการ Run แบบธรรมดา (Normal Mode) จะเป็นการ Run โปรแกรมแบบธรรมดาซึ่งมีการใช้งานแบบปกติในปัจจุบันเป็นการทำงานตามจังหวะเวลาของสัญญาณไฟ ซึ่งเป็นการทำงานตามปกติ
   - ส่วนการ Run แบบ AI (AI Mode) จะเป็นการ Run โปรแกรมแบบพิเศษ มีการคำนวณสัญญาณไฟจราจรก่อนที่จะนำมาใช้ปล่อยรถ โดยคาดหวังว่าจะช่วยลดปัญหาการจราจรติดขัด

ซึ่งการ Run โปรแกรมทั้ง 2 แบบนี้จะนำมาเปรียบเทียบข้อแตกต่างระหว่างกันว่าการคำนวณสัญญาณไฟจราจรแบบ AI กับการตามจังหวะของสัญญาณไฟแบบธรรมดาว่าแตกต่างกันอย่างไรบ้าง เพื่อเป็นการเปรียบเทียบระหว่างจังหวะสัญญาณไฟจราจรแบบธรรมดา กับแบบที่คำนวณได้ว่ามีความแตกต่างกันอย่างไรบ้าง

1.5 ความน่าเชื่อถือของโครงงาน

เนื่องจากโครงงานนี้เป็นโครงงานที่เกี่ยวข้องกับระบบจราจร ดังนั้นจึงต้องอาศัยหลักการของวิศวกรรมการจราจร (Traffic Engineering) มาช่วยในการคำนวณเพื่อให้เกิดความถูกต้องมากที่สุด

แต่เนื่องจากว่าการทำงานของระบบสัญญาณไฟจราจรนั้นมีปัจจัยหลายอย่างมาเกี่ยวข้อง อาจจะเป็นทั้งทางด้านเทคนิค และทางด้านการปฏิบัติ ทางด้านเทคนิคเช่น อัตราการหน่วงเวลาของสัญญาณไฟจราจร หรือการเชื่อมต่อของระบบคอมพิวเตอร์กับสัญญาณไฟจราจรเป็นต้น ส่วนทางด้านการปฏิบัติเช่น การขับรถของบุคคลในถนน (ซึ่งเป็นส่วนสำคัญมากที่ทำให้เกิดข้อผิดพลาดของโปรแกรม) ดังนั้นจึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำลองการทำงานของระบบสัญญาณไฟจราจรบนระบบคอมพิวเตอร์เพื่อทดสอบระบบการทำงาน และเพื่อที่จะควบคุมการทำงานได้ในทุกๆ ด้านก่อนที่จะทำระบบที่ใช้งานได้จริงต่อไป

1.6 วิธีการดำเนินงาน

การดำเนินงานของโครงงานนี้เริ่มด้วยการเก็บรายละเอียดของระบบสัญญาณไฟจราจรที่ใช้ในปัจจุบัน และศึกษาหาปัจจัยต่างๆ ที่มีผลต่อการทำงานของสัญญาณไฟจราจร ซึ่งการหาข้อมูลเกี่ยวกับเรื่องนี้นั้น ในเมืองไทยเองยังไม่มีใครทำงานเรื่องดังกล่าว จึงจำเป็นจะต้องหาข้อมูลจากอินเทอร์เน็ตและโดยการสัมภาษณ์กับบุคคลที่เกี่ยวข้องกับระบบสัญญาณไฟจราจร เมื่อได้รายละเอียดมาแล้วจำเป็นจะต้องหาความเกี่ยวข้องต่างๆ ออกมาเป็นสมการเพื่อใช้ในการทำงานของระบบ รายละเอียดดูได้ในบทที่ 2

จากนั้นก็นำข้อมูลที่ได้มาออกแบบโดยใช้หลักการทางคอมพิวเตอร์มาจัดการทำงานของระบบออกมาเป็นโมเดลการทำงานให้เหมาะสม โดยจะเข้าขั้นตอนของการออกแบบระบบสัญญาณไฟจราจร รายละเอียดอยู่ที่บทที่ 3 เมื่อทำการออกแบบระบบแล้วก็จำเป็นจะต้องพัฒนาโปรแกรมให้สามารถใช้งานได้ การพัฒนาโปรแกรมนั้นจะเป็นในส่วนของบทที่ 4 ซึ่งจะกล่าวถึงรายละเอียดของการสร้างโปรแกรมและหน้าจอของการทำงานของโปรแกรม

เมื่อทำการพัฒนาโครงงานที่ได้แล้วก็จำเป็นจะต้องทำการทดสอบการออกแบบว่าสามารถทำงานได้อย่างถูกต้องหรือไม่ ซึ่งจะทดสอบและสรุปผลการทดสอบในบทที่ 5 และ 6 ในส่วนของบทที่ 6 จะมีการกล่าวถึงแนวทางการพัฒนาโครงงานต่อไปด้วย

อนึ่งการทำงานของระบบนี้บางส่วนเป็นการสมมติขึ้นมา เพราะว่าในการทำงานจริงนั้นได้มีปัจจัยหลายอย่างที่มีผลต่อการทำงาน ผลการทำงานที่ได้ใน โครงงานนี้เมื่อนำไปใช้งานจริงอาจจะไม่เหมือนกับที่ได้ทดลองมาก็เป็นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้พื้นฐาน

2.1 ทฤษฎีทาง Artificial Intelligent (AI)

ทฤษฎีทาง AI เป็นเทคนิคอย่างหนึ่งในการทำให้เครื่องคอมพิวเตอร์มีความสามารถมากขึ้น ซึ่งทฤษฎีทาง AI มีมากมายหลายอย่าง และในโครงงานนี้มีการนำทฤษฎีทาง AI มาใช้เพื่อให้ช่วยคำนวณสัญญาณไฟจราจรให้มีประสิทธิภาพ แต่จะต้องมีหลักการคำนวณที่แน่นอน และหลักการตัดสินใจในกรณีต่างๆ ที่จะเกิดขึ้นอย่างแน่นอน ดังนั้นจึงเป็นหน้าที่ของคนออกแบบโปรแกรมในการอัลกอริทึม (Algorithm) ต่างๆ เพื่อที่จะให้เครื่องคอมพิวเตอร์ฉลาดมากขึ้น และทำงานตามที่ได้ออกแบบไว้อย่างถูกต้องด้วย

เทคนิคทาง AI ที่จะนำมาใช้ในโปรแกรมเพื่อคำนวณหาสัญญาณไฟจราจรนั้นจะต้องเป็นการค้นหาสัญญาณไฟที่ดีที่สุดที่นำมาใช้ในการปล่อยรถให้เกิดประสิทธิภาพสูงสุด โดยที่ประสิทธิภาพของสัญญาณไฟจราจรนั้นสามารถมองได้หลายมุมเช่น มองว่าปล่อยรถให้ได้มากที่สุดในระบบโดยรวม หรือปล่อยรถให้ได้มากที่สุดในถนนที่สนใจโดยกระทบกับถนนด้านอื่นๆ น้อยที่สุดเป็นต้น (รายละเอียดของการทำงานของ AI อัลกอริทึมจะแสดงในหัวข้อที่ 3.2.6 การออกแบบการคำนวณสัญญาณไฟจราจร)

2.2 Object Oriented Design & Object Oriented Programming

Object Oriented Design & Object Oriented Programming (OOD & OOP) เป็นเทคนิคอย่างหนึ่งในการออกแบบโปรแกรมซึ่งมีข้อดีหลายอย่าง คือ

- ทำให้ออกแบบระบบได้ครบถ้วน
- เพิ่มความเข้าใจของปัญหา
- เพิ่มเสถียรภาพในการเปลี่ยนแปลง
- สนับสนุนการนำกลับมาใช้ใหม่
- สนับสนุนการเปลี่ยนขนาดของระบบ
- สนับสนุนการออกแบบที่เชื่อถือได้ และมีความปลอดภัย
- สนับสนุนการทำงานพร้อมกัน

เนื่องจากว่าการออกแบบด้วยวิธีเชิงวัตถุกำลังเป็นที่นิยมในขณะนี้ ดังนั้นภาษาโปรแกรมมิ่งหลายๆ ภาษาจึงได้มีการปรับเปลี่ยนให้เป็นภาษาเชิงวัตถุกันมากยิ่งขึ้น เพราะฉะนั้นโปรแกรมที่พัฒนาขึ้นจึงต้องมีลักษณะของเชิงวัตถุเข้าไปเกี่ยวข้องด้วย ลักษณะของแนวคิดโดยทั่วๆ ไปคือ

1. จะต้องมีคลาส (Class) เป็นเค้าโครงร่างของวัตถุ
2. ในคลาสจะประกอบไปด้วย Attribute ซึ่ง Attribute นี้เป็นคุณลักษณะของวัตถุที่เราสร้างขึ้น แต่ละวัตถุก็จะมี Attribute แตกต่างกันไป
3. Method เป็น Operation ของวัตถุ (หรืออาจจะเรียกว่าเป็นกิริยาของวัตถุก็ได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

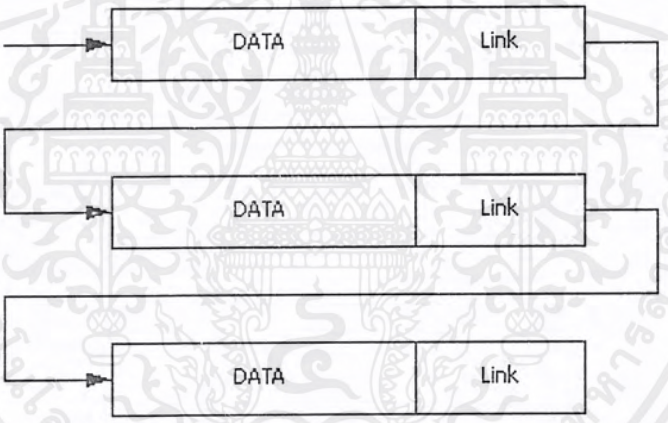
ในโครงงานนี้มีการใช้ภาษา Java ในการพัฒนาโปรแกรม ซึ่งภาษา Java เป็นภาษาที่สนับสนุนกลไกเชิงวัตถุอย่างสมบูรณ์มากที่สุดภาษาหนึ่ง เพราะฉะนั้นในการออกแบบระบบนั้นจะอ้างอิงภาษา Java เป็นหลัก และชนิดของข้อมูลบางชนิดอาจจะเป็นชนิดข้อมูลที่ไม่มีในภาษา Java ก็ได้ แต่เป็นคลาสต้นแบบใน Java Development Kit 1.2.1 ซึ่งสามารถอ้างอิงมาใช้ได้

2.3 Data Structure

Data Structure เป็นการจัดการข้อมูลของโปรแกรม รายละเอียดของ Data Structure ที่ใช้ในโปรแกรมมีดังนี้

2.3.1 Link List

ในโครงงานมีการใช้ Link List ในการเก็บข้อมูลที่เป็นตัวแปรของ Object ในการ Edit โปรแกรม เพราะว่าถ้ามีการกำหนดโครงสร้างข้อมูลแบบ Array เอาไว้จะสิ้นเปลืองเนื้อที่ในหน่วยความจำเป็นอย่างมาก ลักษณะของ Link List เป็นดังรูป



รูปที่ 2-1 แสดงลักษณะของข้อมูลแบบ Link List

จากรูป ลักษณะข้อมูลแบบ Link List จะมีส่วนของตัวข้อมูล และส่วนที่ทำการเชื่อมต่อไปยังตัวแปรตัวต่อไปอยู่ด้วย

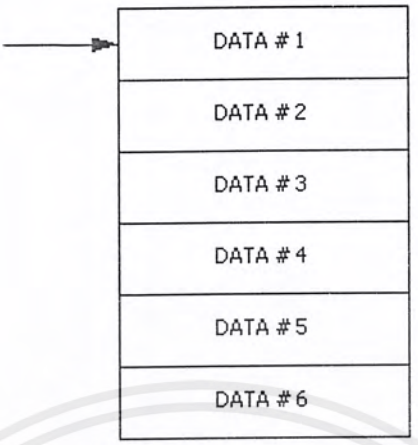
แต่เนื่องจากภาษา JAVA (เป็นภาษาที่ผู้พัฒนานำมาใช้ในการพัฒนา) เองไม่สนับสนุนการทำงานของ Pointer ดังนั้นการทำ Link List จะใช้ในการเชื่อมต่อของวัตถุเข้าด้วยกันนั้น จะต้องมีคลาสมาช่วยจัดการในการแก้ไขปัญหาดังกล่าว (รายละเอียดดูได้ในบทที่ 3 การออกแบบ)

2.3.2 Queue

ในโปรแกรมจะต้องมี Queue มาช่วยในการคำนวณปริมาณรถด้วย เนื่องจากว่ารถจะต้องใช้เวลาในการวิ่งอยู่บนถนนด้วย เพราะฉะนั้นเวลาในการวิ่งจะถูกเก็บเป็นข้อมูลในลักษณะของ Queue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลแบบ Queue จะมีลักษณะคล้ายกับข้อมูลแบบ Link List แต่ลำดับการค้นหาข้อมูลจะเป็นแบบ First-In-First-Out (FIFO) มีลักษณะดังรูป



รูปที่ 2-2 แสดงลักษณะของข้อมูลแบบ Queue

จากรูป เมื่อข้อมูล DATA # 1 ได้ใช้ค่าแล้วจะไม่สามารถอ้างอิงได้อีก ลูกศรจะเลื่อนลงมาอยู่ที่ DATA # 2 แทน และเมื่อข้อมูล DATA # 2 ถูกใช้ไปแล้วลูกศรก็จะชี้ที่ DATA # 3 แทน ในโปรแกรมที่ทำการออกแบบเอาไว้ เมื่ออ้างอิงแล้ว (หรือนำเอาค่ามาใช้แล้ว) ก็จะไม่ใช้ค่านั้นอีก

สำหรับการทำงานในโครงงานนี้จะใช้ Queue ในการเก็บค่าจำนวนรถในถนนเพื่อเป็นการจำลองการทำงานจริงๆ ของถนนนั้นและเพื่อให้ค่าที่ได้จากโครงงานมีค่าใกล้เคียงกับความเป็นจริงมากที่สุด

2.4 Traffic Engineering

จากปัญหาของโปรแกรม เราจะพบว่าจะต้องอาศัยความรู้ทางด้านวิศวกรรมการจราจร (Traffic Engineering) มาช่วย จากการศึกษาของตัวผู้พัฒนาเอง ประกอบกับการออกไปสอบถามข้อมูลจากภายนอก จะได้หัวข้อใหญ่ๆ คือ

2.4.1 ปัจจัยที่มีผลต่อการจราจร

ปัจจัยที่มีผลต่อการจราจร มีหลายปัจจัย แต่พอจะสรุปเป็นปัจจัยหลักๆ ออกเป็น

- ความกว้างของถนน
- ความยาวของถนน
- ความเร็วที่วิ่งได้บนถนน
- สภาพพื้นผิวของถนน
- อุบัติเหตุบนถนน
- สัญญาณไฟจราจร
- ลักษณะการขับรถผู้ขับรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากปัจจัยด้านบน จะเห็นได้ว่ามีส่วนที่เป็นค่า Characteristic ของถนน (สามารถควบคุมได้ ซึ่งได้แก่ ความกว้างของถนน, ความยาวของถนน, ความเร็วที่วิ่งได้บนถนน และสัญญาณไฟจราจร) และส่วนที่เป็น Non-Characteristic ของถนน (ไม่สามารถควบคุมได้ ได้แก่ อุบัติเหตุบนถนน, ลักษณะการขับรถ)

ในส่วนที่เป็นค่า Characteristic ของถนน เราจะสามารถกำหนดลงไปในโปรแกรมได้เลย เพราะว่าเป็นค่าตายตัว ไม่สามารถเปลี่ยนแปลงได้

ส่วน Non-Characteristic ของถนน เป็นส่วนที่เราคาดเดาไม่ได้ แต่สามารถกำหนดให้เกิดเหตุการณ์นั้นๆ ได้ เช่นถ้าเกิดรถชนกันกีดขวางช่องทางจราจร ก็จะทำให้ค่าความกว้างของถนนลดลงไป เป็นต้น

2.4.2 นิยามของรถติด

ปัญหาของโครงงานนี้ (และเป็นปัญหาหลักของโปรแกรม) คือ รถติด ซึ่งเราจะต้องวิเคราะห์สาเหตุของปัญหา เราได้กำหนดนิยามของรถติดว่า "ปริมาณพื้นผิวของถนนที่เหลืออยู่มีจำนวนน้อยกว่าปริมาณพื้นที่ที่รถใช้ไป" ซึ่งเราสามารถแสดงได้ดังรูป

รูปที่ 2-3 แสดงนิยามของรถติดตามที่ได้กำหนดไว้

จากรูปด้านบนจะเห็นได้ว่าปริมาณพื้นผิวที่รถใช้ไปมีปริมาณมากกว่าปริมาณพื้นผิวของถนนที่เหลืออยู่ ซึ่งอาจจะเรียกได้ว่ารถเริ่มจะติดแล้ว สาเหตุของรถติด มีได้หลายสาเหตุ คือ

- ปริมาณรถมีจำนวนเกินพื้นผิวของถนน (พื้นที่ของรถ + พื้นที่ที่รถจอดห่างกัน $\geq$ พื้นผิวของถนน)
- สภาพอากาศ เช่น ในเวลาฝนตกถนนเล็กๆ จะเกิดน้ำท่วมทำให้พื้นผิวของถนนลดลง ดังนั้นรถจึงต้องมาใช้ถนนใหญ่แทน หรือในเวลาหิมะตกจะทำให้พื้นผิวของถนนน้อยลง อีกทั้งยังทำให้ความเร็วของถนนที่วิ่งบนถนนลดลงอีกด้วย (เวลาที่รถระบายออกจากถนน < เวลาที่รถเข้าสู่ถนน จะทำให้ปริมาณรถบนถนนมาก)
- สัญญาณไฟจราจรที่ปล่อยไม่มีประสิทธิภาพ ปล่อยรถในส่วนที่ไม่จำเป็นทำให้ ปริมาณรถที่ระบายออกได้ < ปริมาณรถที่เข้ามาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเกิดอุบัติเหตุต่างๆ เช่น รถชน ไฟไหม้ เป็นต้น ทำให้เสียพื้นผิวจราจร และยังทำให้ความเร็วของรถที่วิ่งได้ลดลงด้วย
- เส้นทางที่รถวิ่งนั้นเป็นเส้นทางเดียว ไม่มีการตัดถนนเพิ่ม จึงทำให้รถมารวมตัวกันมากในถนนเส้นนั้น
- ถนนที่วิ่งผ่านเป็นเส้นทางชุมชน จึงทำให้มีการจอดรถกันที่ริมถนน (เป็นการเสียพื้นผิวการจราจร) และยังจำกัดความเร็วของรถที่วิ่งได้อีกด้วย
- ถนนมีการปรับปรุง เช่น ซ่อมท่อระบายน้ำ วางสายโทรศัพท์ วางท่อบำบัดน้ำเสีย มีการก่อสร้างทาง เป็นต้น ทำให้เสียพื้นผิวการจราจร และยังทำให้ความเร็วของถนนนั้นน้อยอีกด้วย
- ถนนนั้นเป็นถนนที่ผ่านสถานที่สำคัญต่างๆ ทำให้ต้องจำกัดความเร็วในการวิ่งของรถด้วย
- ถนนนั้นเป็นถนนที่รถบรรทุกขนาดใหญ่วิ่ง ทำให้ต้องขับขี่ด้วยความระมัดระวัง และรถบรรทุกเหล่านั้นวิ่งได้ไม่เร็วเท่าที่ควร (เนื่องจากบรรทุกของหนัก) จึงเสมือนกับยึดครองพื้นผิวถนนชั่วคราว
- ถนนนั้นมีทางโค้งมาก ทำให้ความเร็วของรถที่วิ่งบนถนนได้น้อย ซึ่งก็คล้ายกับการยึดครองพื้นผิวถนนไปชั่วคราวด้วย
- สัญญาณไฟจราจรเสีย หรือขัดข้อง ทำให้การควบคุมรถไม่เกิดประสิทธิภาพ
- อยู่ในช่วงเวลาเร่งด่วน ที่ผู้คนต่างมีความต้องการใช้รถ ใช้ถนนอย่างมากจึงเป็นสาเหตุที่ทำให้รถติดได้
- ปริมาณการซื้อรถมีแนวโน้มมากขึ้น เนื่องจากรถถือว่าเป็นสาธารณูปโภคพื้นฐานในปัจจุบัน หรืออาจจะเป็นสิ่งที่บ่งบอกฐานะทางบ้าน จึงทำให้มีปริมาณรถมากขึ้น
- ถนนที่วิ่งได้เป็นถนนทางเดียว (One way) ทำให้รถที่จะวิ่งเข้าไปต้องวิ่งอ้อมไปในทางอื่นเป็นสาเหตุที่ทำให้ปริมาณรถเพิ่มมากขึ้น และเป็นการเสียเวลาในการวิ่งอ้อม ซึ่งอาจจะเรียกได้ว่าเป็นสาเหตุหนึ่งในการเพิ่มปริมาณรถในถนนก็ได้
- ในวันเทศกาลต่างๆ อาจจะมีการจัดงานฉลองรื่นเริงต่างๆ ในบางประเทศมีการจัดแห่ไปตามท้องถนนทำให้รถไม่สามารถวิ่งได้ และถนนบริเวณใกล้เคียงก็จะใช้เป็นพื้นที่จอดรถด้วย ทำให้การจราจรในช่วงนั้นไม่สามารถเคลื่อนที่ได้

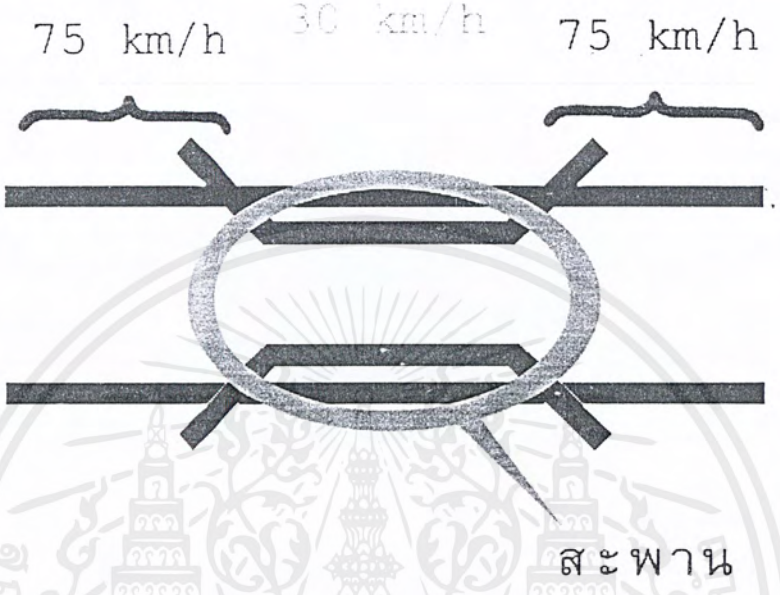
จากการวิเคราะห์สาเหตุดังกล่าว จะพบว่าบางปัญหาไม่สามารถแก้ไขได้ บางปัญหาเป็นสาเหตุทางเทคนิค แต่เราจะพบว่าสาเหตุที่พบได้บ่อยๆ นั้นคือการจัดการด้านสัญญาณไฟจราจรที่ไม่มีประสิทธิภาพ ซึ่งโดยทั่วไปจะปล่อยรถไปตามจังหวะของสัญญาณไฟ ตามเวลาที่ได้กำหนดเอาไว้ ซึ่งควรจะปล่อยตามจำนวนรถมากกว่า จึงเป็นสาเหตุของรถติด

ดังนั้น ถ้าเราใช้การคำนวณสัญญาณไฟจราจรก่อนทำการปล่อยรถ จะเป็นประโยชน์อย่างมาก และจากการทดลองดูแล้วการปล่อยรถให้เป็นจังหวะที่สัมพันธ์กัน จะทำให้ปริมาณรถบนถนนนั้นน้อยลง เป็นการระบายรถไปสู่ถนนที่มีรถติดน้อย ซึ่งการคำนวณนี้จะต้องอาศัยเทคนิค Artificial Intelligent เข้ามาช่วยในการคำนวณสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ความเร็วเฉลี่ยของถนน

ในถนนหนึ่งเป็นไปได้ที่จะมีค่าของความเร็วที่ไม่เท่ากันตลอดทั้งเส้น ดังรูป



รูปที่ 2-4 แสดงลักษณะของถนนที่มีความเร็วไม่เท่ากันทั้งเส้น

จากรูปจะเห็นได้ว่าถนนนั้นมีความเร็วไม่เท่ากัน ดังนั้นเราจึงต้องมีการหาความเร็วเฉลี่ยของถนน ($Speed_{Avg}$) ได้ดังนี้

$$Speed_{Avg} = \sum(Speed_n * Length_n) / \sum(Length)$$

เมื่อ

$Speed_{Avg}$ เป็น ความเร็วเฉลี่ยของถนน (กิโลเมตรต่อชั่วโมง)

$Speed_n$ เป็น ความเร็วของรถในช่วงที่ n (กิโลเมตรต่อชั่วโมง)

$Length_n$ เป็น ความยาวของถนนในช่วงที่ n (เมตร)

$\sum(Length)$ เป็นความยาวของถนนทั้งหมด

แต่ในโปรแกรมได้มีการออกแบบให้หน่วยของความเร็วมีหน่วยเป็น คันต่อวินาทีเลน (car * sec^{-1} * $lane^{-1}$) ซึ่งมีการแปลงดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

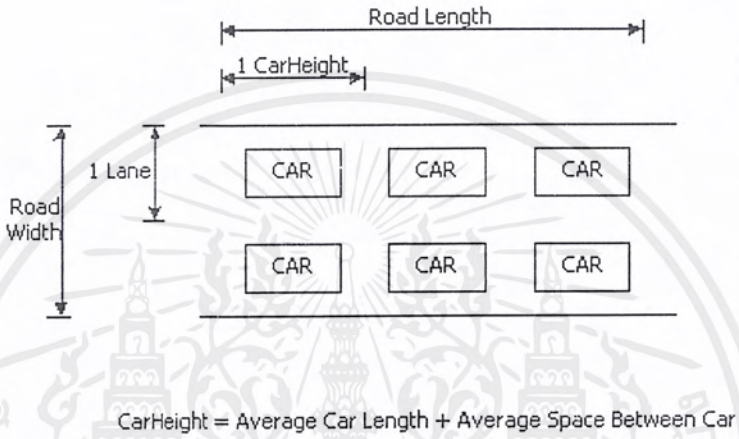
$$Speed_{(car/lane*sec)} = Speed_{(km/h)} * 1000/3600 * 1/CarHeight$$

เมื่อ

$Speed_{(car/lane*sec)}$ เป็น ความเร็วในหน่วยคันต่อวินาทีเลน

$Speed_{(km/h)}$ เป็น ความเร็วในหน่วยกิโลเมตรต่อชั่วโมง

CarHeight เป็น ความยาวของรถโดยเฉลี่ย โดยกำหนดว่ารถ 1 คันมีขนาดเท่ากับ 1 เลน * 1 CarHeight (เมตร) ดังรูป



รูปที่ 2-5 แสดงการหาค่าของ CarHeight

ซึ่งค่า CarHeight จะหาได้จาก

$$CarHeight = Avg(Car\ Length) + Avg(Space\ Between\ Car)$$

เมื่อ

CarHeight เป็น ค่าของความกว้างโดยเฉลี่ยของรถ (เมตร)

Avg(Car Length) เป็น ค่าความกว้างโดยเฉลี่ยของรถ (เมตร)

Avg(Space Between Car) เป็นค่าเฉลี่ยของระยะห่างระหว่างรถที่จอดติดกัน (เมตร)

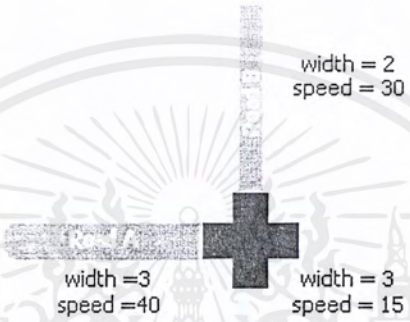
2.4.4 การปล่อยรถที่ทางแยก

การปล่อยรถที่ทางแยกจะมีปัจจัยในการปล่อยรถอยู่หลายปัจจัย ดังนี้

- ความกว้างของเลนที่ปล่อยรถ
- ความกว้างของเลนที่รับรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความกว้างของทางแยก
- ความยาวของระยะทางที่วิ่งบนทางแยก
- ความเร็วของถนนที่ปล่อยรถ
- ความเร็วของถนนที่รับรถ
- ความเร็วของทางแยกที่วิ่งได้
- สภาพพื้นผิวของทางแยก
- เวลาที่ปล่อยรถ



รูปที่ 2-6 แสดงค่าที่เกี่ยวข้องในการเคลื่อนย้ายรถจากถนน A ไปยังถนน B

จากรูปด้านบน เราจะกำหนดไว้ว่า ค่าความยาวของการวิ่งบนทางแยกมีค่าใกล้เคียงกันระหว่างการวิ่งทางตรงกับการวิ่งเลี้ยวซ้าย หรือเลี้ยวขวา และให้สภาพพื้นผิวของทางแยกมีผลโดยตรงกับความเร็วบนทางแยก เราจะได้ว่าปริมาณรถที่ปล่อยได้ในทางแยกจะเป็นได้ ดังนี้

$$CarMove = Min(Speed) * Min(width) * TimeRemain$$

เมื่อ

CarMove เป็น ปริมาณรถที่วิ่งได้ (คัน)

Min(Speed) เป็น ค่าของความเร็วที่น้อยที่สุด (คันต่อวินาทีเลน) ระหว่างถนนที่รถวิ่งเข้าสู่ทางแยก, ถนนที่รถวิ่งออกจากทางแยก และความเร็วในทางแยกเอง

Min(width) เป็น ค่าของความกว้างของถนนที่น้อยที่สุด (เลน) ระหว่างถนนที่รถวิ่งเข้าสู่ทางแยกและถนนที่รถวิ่งออกจากทางแยก

TimeRemain เป็นค่าของเวลาที่ทำการปล่อยรถ (วินาที)

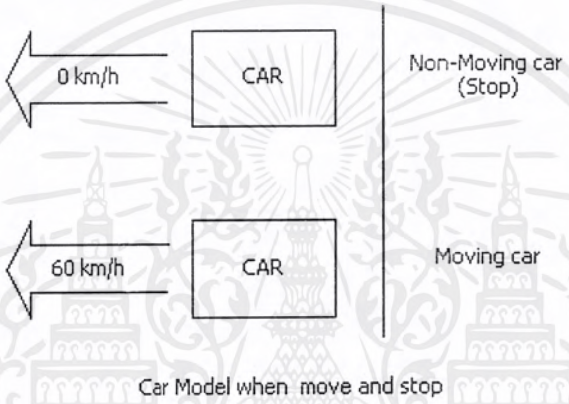
จากสูตรด้านบนเป็นสูตรการคำนวณอย่างง่าย ซึ่งจะมีค่าที่ผิดพลาดเกิดขึ้นได้ (เพราะว่าไม่ได้คิดระยะทางที่รถวิ่งในทางแยก รวมทั้งไม่ได้คิดความเร่งของรถเลย แต่การคำนวณรถที่ทางแยกให้มีความเร่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยจะยากในการคำนวณ และยังมีปัจจัยของการวิ่งอีก ดังนั้นในโปรแกรมจึงเลือกการวิ่งของรถแบบง่ายที่สุด)

2.4.5 ลักษณะการวิ่งของรถ

จากที่กล่าวมาแล้วว่าการวิ่งของรถนั้น จะมีค่าของความเร่งเข้ามาเกี่ยวข้องด้วย แต่ในการคำนวณรถที่มีความเร่งด้วยนั้นจะต้องมีการคำนวณพฤติกรรมของรถทุกคัน (ซึ่งแน่นอนว่ามีความเร่งไม่เท่ากัน) จึงเป็นการยากที่จะคำนวณได้ ดังนั้นจึงได้จำลองการวิ่งของรถให้เป็นแบบที่ง่ายที่สุด ซึ่งการวิ่งของรถที่ง่ายที่สุดคือการวิ่งของรถที่วิ่งดังรูป



รูปที่ 2-7 แสดงลักษณะการจำลองการวิ่งของรถในแบบที่ง่ายที่สุด

จากรูปแสดงหารวิ่งของรถในแบบที่ง่ายที่สุด ซึ่งมีการวิ่งอยู่ 2 แบบคือ

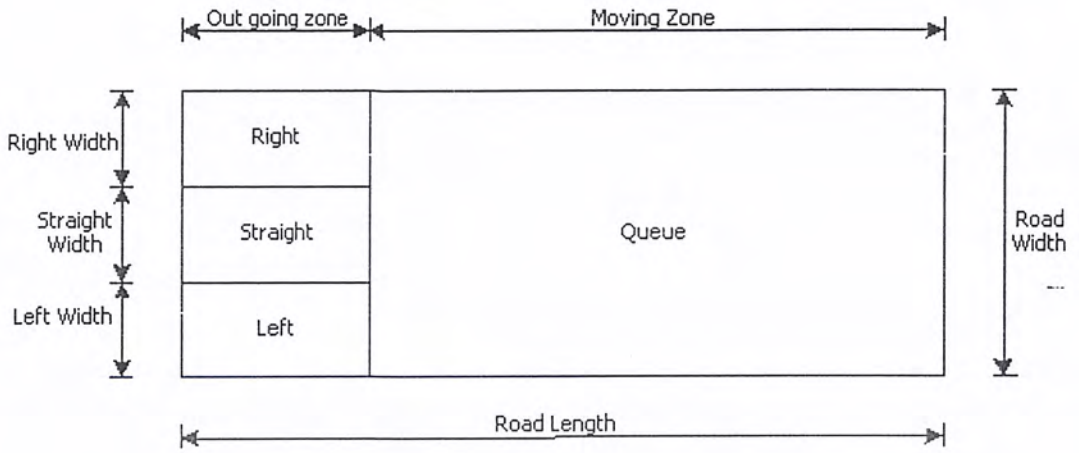
1. เมื่อหยุดนิ่ง (ไม่วิ่ง) จะมีความเร็วเป็น 0 กิโลเมตรต่อชั่วโมงทันที (ไม่มีความหน่วงเลย โดยคิดว่ารถสามารถหยุดได้ทันทีหลังจากที่วิ่ง)
2. เมื่อรถเคลื่อนที่ (วิ่ง) จะมีความเร็วของรถเป็น 60 กิโลเมตรต่อชั่งโมงทันที (คิดว่าไม่มีความเร่งของรถ เมื่อรถวิ่งจะวิ่งได้ด้วยความเร็วจาก 0 กิโลเมตรต่อชั่วโมงเป็น 60 กิโลเมตรต่อชั่งโมงทันที)

จากโมเดลด้านบน จะมีค่าที่ผิดพลาดอยู่บ้าง แต่เพื่อง่ายต่อการคำนวณ จึงจำเป็นจะต้องมีค่าที่ผิดพลาดเกิดขึ้นบ้าง

2.4.6 ลักษณะโดยทั่วไปของถนน

ลักษณะโดยทั่วไปของถนนนั้นจะมีลักษณะดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-8 แสดงลักษณะโมเดลของถนน

จากรูปที่ 2-8 แสดงลักษณะโมเดลของถนน จากการโมเดลดังรูป จะสามารถแบ่งถนนออกเป็น 2 ส่วน คือ

1. ส่วน Out going Zone เป็นส่วนที่รถสามารถเคลื่อนที่ออกจากถนนได้
2. ส่วน Moving Zone เป็นส่วนที่รถอยู่บนถนน ไม่สามารถเคลื่อนที่ออกจากถนนได้ รถจะต้องวิ่งไปยังส่วน Out going Zone เท่านั้น

ซึ่งถนนดังกล่าวจะเป็นถนนแบบ Unit Road คือเป็นถนนที่ไม่มีซอยอยู่เลย การที่ถนนจะเป็นแบบดังรูป เราจะคิดถึงการที่รถกลับรถ (U-turn) ว่ามีค่าน้อยมาก

จากโมเดลเราจะได้ว่า

$$\text{Road Width} = \text{Left Width} + \text{Straight Width} + \text{Right Width}$$

เมื่อ

Road Width คือ ความกว้างของถนน (เลน)

Left Width คือ ความกว้างของเลนที่รถสามารถเลี้ยวไปทางซ้ายได้ (เลน)

Straight Width คือ ความกว้างของเลนที่รถสามารถวิ่งตรงไปได้ (เลน)

Right Width คือ ความกว้างของเลนที่รถสามารถเลี้ยวไปทางขวาได้ (เลน)

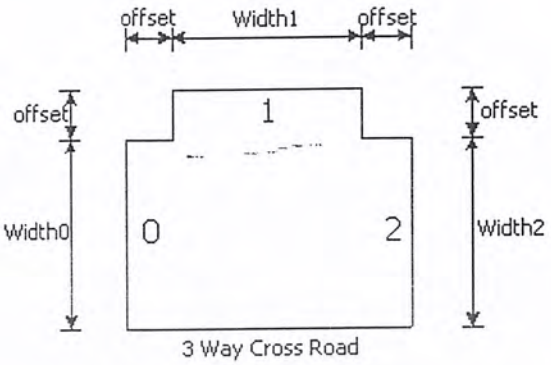
ส่วน Queue ของถนน จะเป็นส่วนที่รถเก็บอยู่ในถนน และมีการเคลื่อนที่ของรถในลักษณะของ Queue เมื่อรถเข้าสู่ Queue และเมื่อรถจะออกจากถนน รถจะออกจากถนนที่ส่วน Left, Straight หรือ Right เท่านั้น

จากลักษณะโดยทั่วไปของถนน เราจะมาดูลักษณะโดยทั่วไปของทางแยกในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.7 ลักษณะโดยทั่วไปของทางแยก

ลักษณะโดยทั่วไปของทางแยก จะเป็นลักษณะที่คล้ายๆ กับของถนน ดังรูป



รูปที่ 2-9 แสดงโมเดลของสามแยก

จากรูปที่ 2-9 จะได้ว่า Width เป็นความกว้างของทางแยก ซึ่งจะได้มาจากความกว้างของถนนที่เชื่อมต่ออยู่ ส่วนค่า offset ถือว่ามีค่าน้อยมากใกล้เคียงศูนย์ ซึ่งถนนในแต่ละด้านละกำหนดให้มีค่าของดรรชนี (Index) ของถนนอยู่ ซึ่งค่าดรรชนีของถนนนี้จะเป็นค่าประจำของทางแยก เพื่อให้อ้างอิงได้ว่าวิ่งจากถนนด้านใดไปด้านใด

สัญญาณไฟจราจรที่ทางแยกจะเป็นสัญญาณไฟจราจรที่บอกว่าทางใดสามารถวิ่งได้ เช่น รถเลี้ยวซ้ายที่ถนนดรรชนีที่ศูนย์ อาจจะเรียกได้อีกอย่างหนึ่งว่า รถจากถนนดรรชนีที่ศูนย์ไปยังถนนดรรชนีที่หนึ่ง เป็นต้น ซึ่งการกำหนดดรรชนีของทางแยกเอาไว้จะสามารถอ้างอิงได้ง่าย และสามารถจัดการในการเขียนโปรแกรมได้ง่ายด้วย

ซึ่งในโครงงานนี้จะมีการอ้างดรรชนีของถนนในการบอกจังหวะของสัญญาณไฟจราจรเพื่อเป็นการง่ายต่อการคำนวณ และเกิดความหมายไม่กำกวมอีกด้วย

2.4.8 สัญญาณไฟจราจร

จากลักษณะของทางแยก เราจะได้ว่าสัญญาณไฟจราจรที่ใช้ในโปรแกรมนั้นเป็นสัญญาณไฟจราจรที่บอกว่ารถสามารถวิ่งจากถนนดรรชนีที่เท่าใดไปยังถนนที่ดรรชนีที่เท่าใด เป็นเวลาเท่าใด ซึ่งโดยปกติแล้วสัญญาณไฟจราจรที่ทางแยกจะมีการกำหนดเอาไว้ตายตัวแล้ว แต่ว่าการจัดการสัญญาณไฟจราจรดังกล่าวจะต้องมีการเปลี่ยนจากปกติมาเป็นสัญญาณไฟจราจรในโปรแกรม

ส่วนสำคัญๆ ของสัญญาณไฟจราจรนั้นจะต้องมี 2 ส่วนคือ

1. ส่วนของจังหวะของสัญญาณไฟจราจร เป็นส่วนที่จะบอกว่ารถจากถนนใดไปสู่ถนนใดได้บ้าง
2. ส่วนของเวลาของสัญญาณไฟจราจรเป็นส่วนที่บอกระยะเวลาในการปล่อยสัญญาณไฟจราจร

ในการคำนวณสัญญาณไฟจราจรจะมีการคำนวณว่าจะเลือก Step ใดในการทำงานและทำงานเป็นเวลาเท่าใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบ (Design Program)

การออกแบบโปรแกรมนี้ จากจุดประสงค์ของโปรแกรมนั้นโปรแกรมที่ทำการพัฒนานั้นเป็นโปรแกรมที่พัฒนาโดยใช้ OOP ดังนั้นการออกแบบโปรแกรมนั้นจะต้องใช้เทคนิค Object Oriented Design ในการออกแบบโปรแกรม (โดยอาศัย Tools ที่มีชื่อว่า Rational Rose 98) ในการช่วยออกแบบคลาส และทำการวิเคราะห์ระบบที่ออกแบบ

3.1 ข้อกำหนดที่ใช้ในการออกแบบ (การสมมติเหตุการณ์ที่เกิดขึ้นในระบบ)

เนื่องจากการออกแบบโปรแกรมนั้นจะต้องกำหนด Specificationion ของโปรแกรมให้ละเอียดว่าจะต้องมีการทำงานเป็นอย่างไรบ้าง มีการกำหนดค่าต่างๆ เป็นอย่างไร หรือกล่าวง่ายๆ ว่าเป็นการกำหนดรายละเอียดของโปรแกรมนั่นเอง

การกำหนดขอบเขตของโปรแกรมในโครงงานนี้ มีการกำหนดรายละเอียดหลายส่วน ซึ่งมีการกำหนดรายละเอียดดังนี้

1. ส่วนของทางแยกในโครงงานจะเป็นทางแยกที่รวมเอา สัญญาณไฟจราจรเข้าไปด้วย และมีการกำหนดรายละเอียดของทางแยกว่าเป็นทางแยกที่สามารถเปลี่ยนจังหวะของสัญญาณไฟได้โดยทันที
2. ส่วนของถนน จะรวมเอารถเข้าไปด้วย โดยรถที่เคลื่อนที่ในถนนจะอาศัยการเคลื่อนที่แบบที่ง่ายที่สุด (สามารถอ่านรายละเอียดได้ที่หัวข้อที่ 2.4.6) โดยรถที่เข้าสู่ถนนจะมีลักษณะเป็นคิว (Queue) ในการเคลื่อนที่ของรถ

3.2 การออกแบบโปรแกรม (Design Program)

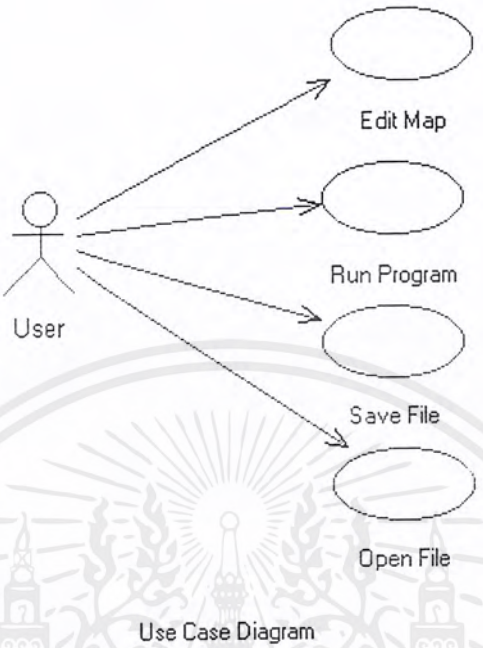
เมื่อเราได้ข้อกำหนดของโครงงานแล้ว ก็จะต้องมาลงรายละเอียดในการทำงานของโปรแกรมว่าผู้ใช้จะสามารถทำงานอะไรได้บ้าง จะมีการออกแบบระบบให้เป็นอย่างไรมีการทำงานเป็นอย่างไรเป็นต้น ซึ่งรายละเอียดต่างๆ ดังที่ได้กล่าวมาแล้วนั้นจะอาศัยการออกแบบแบบ Object Oriented Design, OOD ในการออกแบบดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.2.1 Use case diagram

Use case diagram ของโปรแกรมนั้น เราสามารถออกแบบได้ดังรูป



รูปที่ 3-1 แสดง Use Case Diagram

จากรูปด้านบนแสดงลักษณะของ Use Case Diagram ที่ผู้ใช้ (User) สามารถสร้างแผนที่, Run โปรแกรม, บันทึกข้อมูล และเปิดอ่านข้อมูลได้ โดยเราสามารถจำแนกรายละเอียดของโปรแกรมย่อยลงไปได้อีก ซึ่งจะสามารถแสดงได้ในการออกแบบคลาสของโปรแกรม

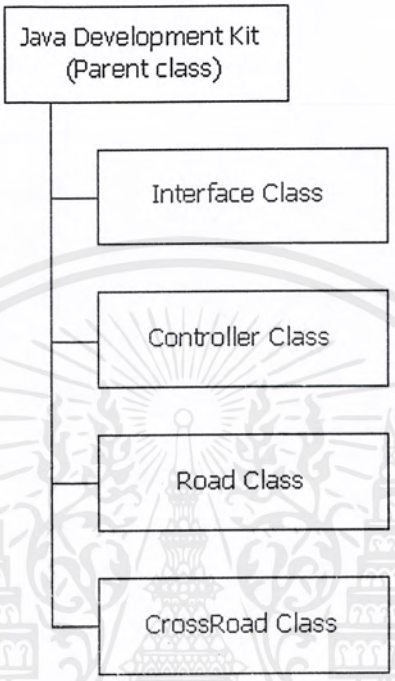
3.2.2 การออกแบบคลาส (Class Design)

การออกแบบคลาสของโปรแกรมนี้จะเป็นการออกแบบที่นับเอาเฉพาะคลาสหลักๆ ของโปรแกรมเท่านั้น เพราะว่าการเขียนโปรแกรมนั้นจะมีคลาสที่เป็นส่วนของ Interface อีกเยอะมาก (ซึ่งในที่นี้ไม่ได้อธิบายในส่วนนั้น)

หลักการในการออกแบบนั้น จากหลักการออกแบบ OOD เราจะต้องกำหนดคลาสที่มีตัวตนขึ้นมาก่อน จากปัญหาของโปรแกรมนั้นจะต้องกำหนดคลาสหลักๆ ของโปรแกรมคือ คลาสทางแยก คลาสถนน คลาสของรถ แต่เนื่องจากคลาสของรถนั้นในโปรแกรมจะต้องสร้างวัตถุจำนวนมากขึ้นมา และในการมองหลักการทำการจราจรนั้นไม่จำเป็นจะต้องทราบว่ารถแต่ละคันจะไปทางใด และจะไปด้วยวิธีอย่างไร เราเพียงแต่สนใจว่ารถนั้นอยู่บนถนนมีปริมาณเท่าไร และจะต้องจัดการกับรถดังกล่าวอย่างไรจึงจะให้เกิดรถติดน้อยที่สุด ด้วยเหตุผลดังกล่าวเราจึงออกแบบให้คลาสของรถนั้นไม่มี แต่เนื่องจากว่าการทำงานของทางแยกทุกๆ ทางแยกจะต้องสามารถเชื่อมโยงถึงกัน ดังนั้นเราจึงจำเป็นจะต้องสร้างคลาสขึ้นมาอีก 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
42820

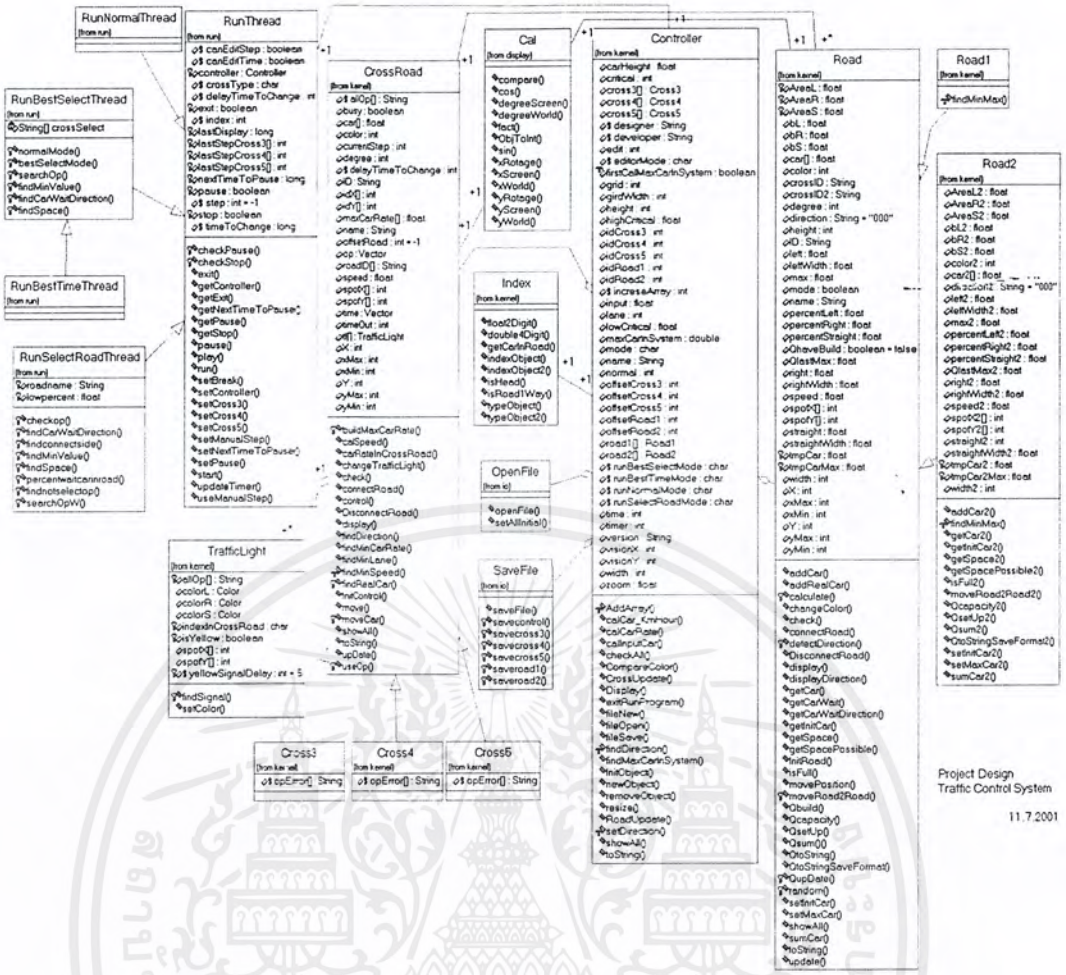
คลาสเพื่อเป็นคลาสที่จัดการเกี่ยวกับการทำ AI ทั้งหมด เราจึงได้ทำการออกแบบคลาสขึ้นมาเป็นคลาส Controller เพื่อเป็นคลาสที่จัดการเกี่ยวกับการควบคุมสัญญาณไฟจราจรทั้งหมด ส่วน Interface ของโปรแกรมนั้นเป็นส่วนที่จะสร้างขึ้นมาโดย Interface ดังกล่าวจะต้องยึดติดกับภาษาที่ใช้เขียนเป็นอย่างมาก ดังนั้นเราจึงไม่กล่าวไว้ในการออกแบบ ดังนั้นเราจะได้การออกแบบคลาสอย่างคร่าวๆ เป็นดังรูป



รูปที่ 3-2 แสดงภาพรวมของการออกแบบคลาส

เมื่อเราได้โครงร่างของการออกแบบอย่างคร่าวๆ แล้ว ขั้นตอนต่อไปเราจะต้องทำการออกแบบรายละเอียดของคลาสแต่ละคลาสอย่างละเอียด เพื่อที่จะทำการเขียนเป็นโปรแกรมต่อไป (การออกแบบคลาสเราจะใช้ Tools ที่มีชื่อว่า Rational Rose 98 ในการเขียน และทำการคัดลอกหน้าจอมาแสดงผลในรายงานฉบับนี้) การออกแบบแผนภาพของคลาส (Class Diagram) นั้นมีลักษณะดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-3 แสดงลักษณะของคลาสที่ได้ทำการออกแบบไว้ในมาตรฐานของ UML

จากรูปด้านบนเป็นการออกแบบคลาสที่สมบูรณ์ (รายละเอียดต่างๆ จะอธิบายในหัวข้อถัดไป) การออกแบบคลาสด้านบนอาจจะดูยากเนื่องจากมีการอ้างอิงมาตรฐานของ UML และเราจะแสดงรายละเอียดของคลาสต่างๆ ได้โดยจะอธิบายใน Data Dictionary โดยละเอียดในหัวข้อถัดไป

3.2.3 Data Dictionary

เป็นส่วนของการแสดงรายละเอียดของการออกแบบโปรแกรม เอาไว้อย่างละเอียด ซึ่งเมื่อทำการออกแบบรายละเอียดเอาไว้ในโปรแกรมแล้วจะต้องมีการเขียนรายละเอียดของโปรแกรมเอาไว้ใน Data Dictionary ด้วยเพื่อให้การอ่านการออกแบบของโปรแกรมทำได้อย่างถูกต้อง และเป็นที่เข้าใจกันระหว่างผู้พัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.1 คลาส Controller

เป็นคลาสหลักที่ทำหน้าที่ควบคุมโปรแกรม อีกทั้งยังเป็นส่วนที่เก็บค่าวัตถุของถนนและทางแยกอีกด้วย การทำงานของโปรแกรมทั้งหมดจะต้องทำงานผ่านที่คลาสนี้ ไม่ว่าจะเป็นการเปิดแผนที่ บันทึกแผนที่ สร้างแผนที่ แก้ไขแผนที่ และการจำลองการทำงานของแผนที่ มีลักษณะของวัตถุ สามารถแสดงได้ในตารางดังนี้

ข้อมูล	ชนิดของข้อมูล	รายละเอียด
carHeight	float	ขนาดความยาวของรถ มีหน่วยเป็น เมตร/คัน (อ่านรายละเอียดได้ในหัวข้อ 2.4.3 รูปที่ 2-5)
critical	int	สีที่แสดงสถานะของถนนในสถานะ critical (ดังรูปที่ 3-4)
cross3[]	Cross3	อาร์เรย์ของวัตถุสามแยก
cross4[]	Cross4	อาร์เรย์ของวัตถุสี่แยก
cross5[]	Cross5	อาร์เรย์ของวัตถุห้าแยก
designer	String	ชื่อของผู้ออกแบบระบบ
developer	String	ชื่อของผู้พัฒนาระบบ
editorMode	char	โหมดการทำงานแบบการสร้างแผนที่ (ดูรายละเอียดในส่วนของ Attribute ของ mode)
firstCalMaxCarInSystem	boolean	Flag ที่บอกว่ามีการคำนวณจำนวนรถรวมของระบบหรือไม่ (ใช้ในตอนการ Initial Run โปรแกรม)
grid	int	ขนาดของ grid ที่จะแสดงผลที่หน้าจอว่ามีขนาดเท่าใด มีหน่วยเป็น จุด
gridWidth	int	ขนาดของความยาวถนนต่อ 1 grid มีหน่วยเป็น เมตร/grid
height	int	ขนาดความสูงของแผนที่ของโปรแกรม มีหน่วยเป็น จุด (ดูรายละเอียดเพิ่มเติมได้ที่หัวข้อ 3.2.3.13 คลาส Cal)
highCritical	float	เปอร์เซ็นต์สูงสุดที่ถนนยังเป็น critical มีหน่วยเป็นเปอร์เซ็นต์ของรถในถนน (ดูรูปที่ 3-4)
idCross3	int	ค่าที่ใช้ในการสร้าง ID ของสามแยกให้ไม่ซ้ำกัน
idCross4	int	ค่าที่ใช้ในการสร้าง ID ของสี่แยกให้ไม่ซ้ำกัน
idCross5	int	ค่าที่ใช้ในการสร้าง ID ของห้าแยกให้ไม่ซ้ำกัน
idRoad1	int	ค่าที่ใช้ในการสร้าง ID ของถนนทางเดียวให้ไม่ซ้ำกัน
idRoad2	int	ค่าที่ใช้ในการสร้าง ID ของถนนสองทางให้ไม่ซ้ำกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

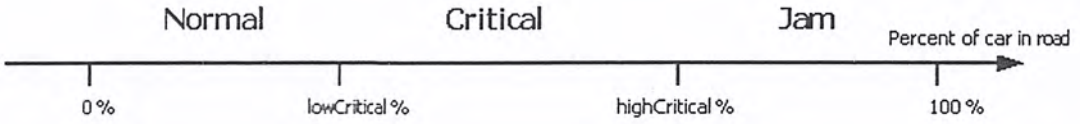
increseArray	int	ค่าคงที่ที่ใช้ในการเพิ่มค่าของอาร์เรย์ของวัตถุในโปรแกรม (วัตถุในโปรแกรมได้แก่ สามแยก สี่แยก ห้าแยก ถนนทางเดียว ถนนสองทาง)
input	float	ค่าที่ใช้ในการเพิ่มค่าเข้าสู่ถนน มีหน่วยเป็น เปอร์เซ็นต์ของจำนวนรถมากที่สุดที่ถนนรับได้ หรือจากสูตร Input Car = input * max car in that road / 100 เมื่อ Input Car = จำนวนรถที่เข้าสู่ถนนนั้น input = เปอร์เซ็นต์ของรถที่เข้าสู่ถนน max car in that road = จำนวนรถสูงสุดที่ถนนนั้นรับได้
lane	int	จำนวนเลนต่อ grid มีหน่วยเป็น grid/เลน
lowCritical	float	ค่าต่ำสุดของ Critical ของถนน มีหน่วยเป็น เปอร์เซ็นต์ของปริมาณรถในถนน (ดูรายละเอียดได้จากรูปที่ 3-4)
maxCarInSystem	double	ปริมาณรถในถนนทั้งหมดของระบบที่สามารถรับได้
mode	char	โหมดที่ใช้ในการทำงานของโปรแกรม มีรายละเอียดคือ • Controller.editorMode • Controller.runNormalMode • Controller.runSelectRoadMode • Controller.runBestSelectMode • Controller.runBestTimeMode (รายละเอียดของโหมดการทำงานทั้งหมดอ่านได้ตามตารางด้านล่าง โดยจะมีการอ้างอิงรายละเอียดของ Attribute เป็นแบบ Static Variable เพื่อการใช้งานที่ง่าย และเหมือนกันทั้งระบบ)
name	String	ชื่อของแผนที่ที่ได้สร้างขึ้น
offsetCross3	int	ค่าที่ใช้บอกปริมาณของสามแยกที่อยู่ในแผนที่
offsetCross4	int	ค่าที่ใช้บอกปริมาณของสี่แยกที่อยู่ในแผนที่
offsetCross5	int	ค่าที่ใช้บอกปริมาณของห้าแยกที่อยู่ในแผนที่
offsetRoad1	int	ค่าที่ใช้บอกปริมาณของถนนทางเดียวที่อยู่ในแผนที่
offsetRoad2	int	ค่าที่ใช้บอกปริมาณของถนนสองทางที่อยู่ในแผนที่
road1[]	Road1	อาร์เรย์ของถนนทางเดียว
road2[]	Road2	อาร์เรย์ของถนนสองทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

runBestSelectMode	char	โหมดการทำงานแบบ จำลองการทำงานของสัญญาณไฟจราจรแบบปล่อยปริมาณรถมากที่สุด (คำนวณเวลาและเลือกสัญญาณไฟ)
runBestTimeMode	char	โหมดการทำงานแบบ จำลองการทำงานของสัญญาณไฟจราจรแบบปล่อยเวลาของการทำงานของสัญญาณไฟจราจรให้คุ้มค่าที่สุด โดยจะมีการคำนวณเวลาในการปล่อยสัญญาณไฟจราจร (ใช้การคำนวณเวลาอย่างเดียว)
runNormalMode	char	โหมดการทำงานแบบ จำลองการทำงานของสัญญาณไฟจราจรแบบปรกติ (ตั้งสัญญาณไฟแบบอัตโนมัติ)
runSelectRoadMode	char	โหมดการทำงานแบบ ปล่อยรถในถนนที่สนใจออกให้ได้มากที่สุด (ใช้การคำนวณเวลาและเลือกสัญญาณไฟที่ดีที่สุด เพื่อปล่อยรถให้คุ้มค่าที่สุด)
time	long	เวลาที่ใช้ในการจำลองการทำงานของสัญญาณไฟจราจร มีหน่วยเป็นวินาที และเมื่อจำลองการทำงานแบบไม่มีสิ้นสุดจะมีค่าเท่ากับ –1
timer	long	เวลาปัจจุบันที่โปรแกรมได้จำลองการทำงาน มีหน่วยเป็นวินาที โดยค่าจะเพิ่มขึ้นเรื่อยๆ และค่าจะไม่เท่ากับเวลาจริง (เป็นการจำลองการทำงาน ไม่ได้เป็นการทำงานแบบ Real time)
version	String	เวอร์ชันของระบบสัญญาณไฟจราจร
visionX	int	มุมมองของแผนที่ในแกน X ของ World Coordinate (ดูรายละเอียดได้ที่หัวข้อ 3.2.3.13 คลาส Cal)
visionY	int	มุมมองของแผนที่ในแกน Y ของ World Coordinate (ดูรายละเอียดได้ที่หัวข้อ 3.2.3.13 คลาส Cal)
width	int	ความกว้างของแผนที่ มีหน่วยเป็นจุด (ดูรายละเอียดได้ที่หัวข้อ 3.2.3.13 คลาส Cal)
zoom	float	ขนาดของจำนวนเท่าที่ใช้ในการแสดงผลของโปรแกรม (ดูรายละเอียดได้ที่หัวข้อ 3.2.3.13 คลาส Cal)

ตารางที่ 3-1 แสดงรายละเอียดของ Attribute ของคลาส Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 แสดงการกำหนดการแสดงผลของถนนว่าเป็นเช่นใด

จากรูปเราจะแบ่งสถานะของถนนออกเป็น 3 ประเภทใหญ่ๆ คือ

- Normal (ปรกติ) หมายความว่าถนนนั้นมีปริมาณรถที่อยู่บนถนนน้อยมาก ซึ่งจะได้ว่าอัตราการวิ่งของรถ (Car Rate) นั้นมีมาก หรือพูดง่ายๆ ว่ารถไม่ติดนั่นเอง
- Critical หมายความว่าถนนนั้นมีปริมาณรถเริ่มจะมากแล้ว แต่รถยังสามารถวิ่งได้อยู่ แต่ว่าถ้ารับปริมาณรถมากขึ้นจะทำให้รถติดได้ หรือพูดง่ายๆ ว่ารถเริ่มจะติดแล้ว
- Jam หมายความว่าถนนนั้นมีปริมาณรถอยู่หนาแน่นมาก รถเคลื่อนที่ได้ช้ามาก ถนนนั้นอาจจะรับปริมาณรถได้เพียงเล็กน้อยหรืออาจจะไม่ได้เลย หรือพูดง่ายๆ ว่ารถติดแล้ว (อาจจะเป็นติดแบบธรรมดา หรือติดแบบหนักก็ได้ แล้วแต่ผู้ใช้กำหนด)

จาก Data Dictionary ด้านบน ยังไม่ได้อธิบายการทำงานของ Method ในคลาส Controller เลย รายละเอียดของ Method ของคลาส Controller มีดังนี้

ชนิดของ Method	รายละเอียด
void	addArray(String object) ใช้สำหรับขยายขนาดของอาร์เรย์ของวัตถุให้มีขนาดใหญ่ขึ้น เป็นขนาด increseArray
public float	calCar_KmHour(float speed_car_s) ใช้สำหรับแปลงค่าความเร็วในหน่วย car/(sec*lane) ไปเป็นความเร็วในหน่วย km/hour
public float	calCarRate(float speed_Km_h) ใช้สำหรับแปลงค่าความเร็วในหน่วย km/hour ไปเป็นความเร็วในหน่วย car/(sec*lane)
public float	calInputCar(float maxSpacePossible) ใช้สำหรับคำนวณปริมาณรถจากระบบภายนอกเข้าสู่ถนน โดยส่งค่ากลับไปเป็นปริมาณรถที่เข้าสู่ถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public boolean	checkAll() ใช้สำหรับตรวจสอบระบบทั้งหมดก่อนว่ามี Error เกิดขึ้นหรือไม่ก่อนการจำลองการทำงานของสัญญาณไฟจราจร โดยถ้า return true ก็คือระบบที่สร้างนั้นไม่มีปัญหา
public static Color	CompareColor(int color) ใช้สำหรับแปลงค่าจากสีของวัตถุมาเป็นสีที่สามารถใช้แสดงผลได้จริงในโปรแกรม
public void	CrossUpdate() ใช้สำหรับการ update ทางแยกทุกทางแยก เรียกใช้เมื่อจำลองการทำงานของโปรแกรม
public void	Display() ใช้สำหรับการแสดงผลคำนวณจุดการแสดงผลของวัตถุในระบบทั้งหมด
public synchronized void	exitRunProgram() ใช้เมื่อสิ้นสุดการจำลองระบบสัญญาณไฟจราจรทั้งหมดของระบบและเปลี่ยนโหมดการทำงานมาเป็น editorMode
public void	fileNew(int width, int height) ใช้เมื่อเริ่มต้นการสร้างระบบสัญญาณไฟจราจร โดยใส่ค่าความกว้างและความสูงของแผนที่ในระบบ
public void	fileOpen(String fileName, String filePath) ใช้เพื่อเปิดไฟล์ที่ได้เคยบันทึกเอาไว้แล้วมาใช้
public void	fileSave(String fileName, String filePath) ใช้สำหรับบันทึกแผนที่เอาไว้เป็นไฟล์ชนิด Text
String	findDirection() ใช้สำหรับหาทิศทางของถนนที่สามารถวิ่งไปได้ ถูกเรียกใช้จาก setDirection() อีกทีหนึ่ง
public void	findMaxCarInSystem() ใช้สำหรับหาปริมาณรถสูงสุดที่ถนนทั้งหมดในระบบสามารถรับค่าได้
public void	InitObject() ใช้สำหรับการกำหนดค่าต่างๆ ก่อนที่จะจำลองการทำงานของระบบสัญญาณไฟจราจร
public void	newObject(String object, int x, int y, int degree) ใช้สำหรับการสร้างวัตถุในระบบ โดยมีค่าตำแหน่งแกน X แกน Y และองศาตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public void	removeObject(String object, int index) ใช้สำหรับลบวัตถุออกจากระบบ ที่ตำแหน่ง index ในอาร์เรย์ของวัตถุนั้นๆ
public void	resize(int width, int height) ใช้สำหรับเปลี่ยนขนาดของแผนที่ในระบบ
public void	RoadUpdate() ใช้สำหรับ Update ถนนทั้งหมดในระบบ ในโหมดของการจำลองการทำงานของสัญญาณไฟจราจร
public void	setCarHeight(float carHeight) ใช้กำหนดขนาดของ carHeight ของระบบ (ค่าของ carHeight ดูได้จากหัวข้อที่ 2.4.3)
void	setDirection() ใช้สำหรับการกำหนด direction ของถนนทั้งหมดก่อนการจำลองการทำงานของสัญญาณไฟจราจร ถูกเรียกใช้โดย InitObject()
public void	showAll() ใช้สำหรับแสดงค่า Attribute ต่างๆ ของระบบทั้งหมดออกทางหน้าจอ (แสดงผลออกทาง command line ใน DOS)
public String	toString() ใช้สำหรับแสดงค่า Attribute ต่างๆ ของระบบออกเป็น String

ตารางที่ 3-2 แสดงรายละเอียดของ Method ของคลาส Controller

3.2.3.2 คลาส **CrossRoad** (คลาสทางแยก)

เป็นคลาสของทางแยก โดยจะเป็นคลาสต้นแบบของคลาสทางแยกทั้ง 3 แบบที่มีในการออกแบบ คือ คลาส Cross3, คลาส Cross4 และคลาส Cross5 จะมี Attribute ต่างๆ ของวัตถุเป็นดังตารางดังนี้

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
allOp[]	String	เป็น op ทั้งหมดที่ทางแยกสามารถใช้งานได้ ซึ่งเป็นค่าคงที่มีค่าเป็นดังนี้ {"01", "02", "03", "04", "10", "12", "13", "14", "20", "21", "23", "24", "30", "31", "32", "34", "40", "41", "42", "43"}
busy	boolean	Flag แสดงสถานะของทางแยกว่าสามารถสั่งงานได้หรือไม่ ถ้ามีค่าเป็น True แสดงว่าทางแยกไม่ว่าง ไม่สามารถสั่งงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

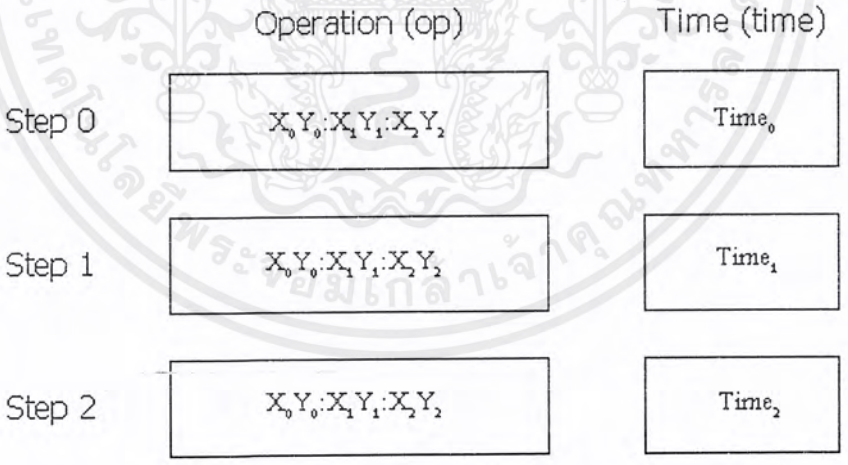
car[]	float	เป็นปริมาณรถที่ปล่อยไปแล้วในทางแยก โดยจะเก็บค่าเอาไว้เพื่อใช้ในการปัดเศษของปริมาณรถที่วิ่งให้เป็นจำนวนเต็ม
color	int	สีของทางแยก
currentStep	int	จังหวะของสัญญาณไฟที่ทางแยกปล่อยอยู่
degree	int	องศาของทางแยก (ใช้ในส่วนของการแสดงผลทางแยก)
delayTimeToChange	int	เวลาของสัญญาณไฟเหลือง มีหน่วยเป็นวินาที โดยมีค่าในระบบเป็น 5 วินาที
ID	String	Id ของทางแยก เพื่อใช้ในการระบุวัตถุทางแยกในโปรแกรม
idX[]	int	จุดที่ใช้แสดงผลด้านของทางแยกที่ได้กำหนดเอาไว้ในระบบให้ผู้ใช้สามารถมองเห็น และกำหนดด้านของทางแยกได้อย่างถูกต้อง โดยเป็นจุดในแกน X
idY[]	int	จุดที่ใช้แสดงผลด้านของทางแยกที่ได้กำหนดเอาไว้ในระบบให้ผู้ใช้สามารถมองเห็น และกำหนดด้านของทางแยกได้อย่างถูกต้อง โดยเป็นจุดในแกน Y
maxCarRate[]	float	ค่า car rate สูงสุดที่รถสามารถวิ่งได้ โดยจะคำนวณเมื่อตอนเริ่มสั่งการปล่อยรถ เนื่องจากว่าความสามารถในการวิ่งของรถผ่านทางแยกนั้นมีหลายปัจจัย จึงต้องคำนวณค่าสูงสุดที่เป็นไปได้ในการวิ่งของรถ (อ่านรายละเอียดได้จากหัวข้อที่ 2.4.4 โดยที่ค่า Time Remain มีค่าเป็น 1)
name	String	ชื่อของทางแยกในระบบ ควรจะตั้งชื่อให้ไม่ซ้ำกันเพื่อความง่ายในการอ้างอิงถนน และความถูกต้องของระบบ
offsetRoad	int	ค่าที่ใช้บอกถนนที่มาเชื่อมต่อกับทางแยกว่ามีการเชื่อมต่อของถนนครบหรือไม่ โดยจะมีค่าเริ่มต้นเป็น –1
op	Vector of String	จังหวะของสัญญาณไฟจราจรในทางแยกนั้นๆ (ดูที่รูป 3-5)
roadID[]	String	Id ของทางแยกที่ใช้บอกว่าถนนใดเชื่อมต่ออยู่ โดยมีการเก็บด้านที่ถนนเชื่อมต่ออยู่ด้วย มีค่าเป็นดังนี้ Id ของถนน + “H”/”F” เมื่อ “H” หมายถึง เชื่อมต่อกับถนนที่ด้านหัว (Head) “F” หมายถึง เชื่อมต่อกับถนนที่ด้านปลาย (Foot)
speed	float	ค่าความเร็วที่รถวิ่งได้บนทางแยก มีหน่วยเป็น car/(lane*sec)
spotX[]	int	จุดที่ใช้ในการแสดงผลของทางแยก ในแกน X
spotY[]	int	จุดที่ใช้ในการแสดงผลของทางแยกในแกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

time	Vector of Integer	เวลาที่ใช้ในการตั้งจังหวะของสัญญาณไฟจราจร มีหน่วยเป็นวินาที (ดูรูปที่ 3-5)
timeOut	long	เวลาที่หยุดการปล่อยสัญญาณไฟของ currentStep มีหน่วยเป็นวินาที
tl[]	TrafficLight	วัตถุของสัญญาณไฟจราจร (ดูรายละเอียดที่หัวข้อ 3.2.3.9 คลาส TrafficLight)
X	int	ตำแหน่งของวัตถุในแกน X (ใช้ในการแสดงผลวัตถุ)
xMax	int	ค่า X มากที่สุดที่วัตถุมี (ใช้ในการตรวจหาวัตถุ)
xMin	int	ค่า X น้อยที่สุดที่วัตถุมี (ใช้ในการตรวจหาวัตถุ)
Y	int	ตำแหน่งของวัตถุในแกน Y (ใช้ในการแสดงผล)
yMax	int	ค่า Y มากที่สุดที่วัตถุมี (ใช้ในการตรวจหาวัตถุ)
yMin	int	ค่า Y น้อยที่สุดที่วัตถุมี (ใช้ในการตรวจหาวัตถุ)

ตารางที่ 3-3 แสดงรายละเอียดของ Attribute ของคลาสทางแยก

จากตารางด้านบนเป็นการแสดง Attribute ของทางแยกอย่างคร่าวๆ เอาไว้ โดยลักษณะของ op ที่เก็บและใช้ในระบบจะมีค่าเป็นดังรูป



รูปที่ 3-5 แสดงลักษณะของ op และ time ที่ใช้ในทางแยก

จากรูปจะเห็นว่ามีส่วนของ op และ time ที่มีขนาดเท่ากัน โดย op เป็น Vector of String (คลาส Vector เป็นคลาสของ JAVA ใช้ในการสร้างอาร์เรย์ขนาดไม่จำกัด เป็นอาร์เรย์ที่สามารถเพิ่มค่าขึ้นได้เรื่อยๆ) และ time เป็น Vector of Integer โดย op มีลักษณะเป็นค่า "จากถนนที่ index X_0 ไปยังถนนที่ index Y_0" เช่น 01 หมายถึงรถวิ่งจากถนนที่เชื่อมต่อทางแยกที่ index 0 ไปยังถนนที่เชื่อมต่อทางแยกที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

index 1 เป็นต้น ส่วนเครื่องหมาย ":" ใช้สำหรับแยกจังหวะของสัญญาณไฟจราจรของทางแยกออกจากกัน เวลาที่กำหนด step การทำงานของทางแยกจะเก็บไว้ใน time โดย time มีค่าเป็นวินาที

การกำหนด op ของทางแยกนั้นสำคัญมาก เพราะว่าทางแยกแต่ละทางแยกนั้นไม่จำเป็นจะต้องมี op ที่เท่ากัน และถนนที่เชื่อมต่อทางแยกนั้นก็ไม่เหมือนกันด้วย การทำงานของระบบสัญญาณไฟจราจรจะมีประสิทธิภาพมากเท่าใด ก็ขึ้นอยู่กับ op ที่ตั้งเอาไว้ด้วย

Method ที่ใช้ในทางแยกนั้น บาง Method เป็น abstract Method โดยทำการกำหนดรายละเอียดไว้ในคลาสลูกของทางแยกทีหลัง รายละเอียดของ Method แสดงได้ในตาราง

ชื่อของ Method	รายละเอียด
public void protected void	buildMaxCarRate(Controller controller) buildMaxCarRate(int counter, String Op, Controller controller) ใช้สำหรับหาค่าของ maxCarRate ของทางแยก โดยที่ Method แรกเป็นการเรียกใช้จากการเปิดไฟล์ขึ้นมาทำงานต่อ ส่วนในอีก Method หนึ่งเป็นการเรียกใช้งานภายในในระหว่างจำลองการทำงานของระบบ
public void	calSpeed(float speed_km_h, Controller controller) ใช้สำหรับคำนวณความเร็วในทางแยก
public float	CarRateInCrossRoad(int speec_km_h, Controller controller) ใช้แปลงค่าจากความเร็วในหน่วย Km/h เป็นความเร็วในหน่วยของ คัน/(วินาที*เลน)
abstract public boolean	ChangeTrafficLight(Controller controller) ใช้สำหรับเปลี่ยนค่าสัญญาณไฟจราจร ถ้า return true แสดงว่าสัญญาณไฟมีการเปลี่ยนแปลง
abstract public boolean	check(Controller controller) ใช้สำหรับตรวจสอบว่าทางแยกนั้นสามารถจำลองการทำงานได้หรือไม่ ถ้า return true หมายความว่าทางแยกนั้นไม่มีข้อผิดพลาด
public void	connectRoad(String roadID, int offset, Controller controller) ใช้สำหรับเชื่อมต่อถนนกับทางแยกเข้าด้วยกันที่ตำแหน่ง index ของถนนที่ทางแยกเท่ากับ offset (ดูรูปที่ 3-6)
public synchronized void	control(int indexOp, long time, Controller controller) ใช้สำหรับสั่งงานทางแยกให้ทำการปล่อยสัญญาณไฟจราจรที่จังหวะเป็น indexOp และเป็นเวลา time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public void	DisconnectRoad(int offset, Controller controller) ใช้สำหรับตัดการเชื่อมต่อของถนนกับทางแยกที่ตำแหน่ง index ของถนนที่ทางแยกเป็น offset
abstract public void	display() ใช้สำหรับแสดงผลทางแยก โดยการคำนวณค่าจุดต่างๆ ภายในวัตถุ
abstract public char	findDirection(int indexRoadIn, int indexRoadOut) ใช้สำหรับหาค่าทิศทางที่รถวิ่ง จาก index ของถนนเท่ากับ indexRoadIn ไปยังถนนที่ index เท่ากับ indexRoadOut โดยมีการ return ค่ากลับมาเป็นทิศทาง คือ 'L', 'S' และ 'R'
public float	findMinCarRate(int indexRoadIn, int indexRoadOut, Controller controller) ใช้สำหรับหาค่า Car Rate ที่น้อยที่สุดที่ทางแยกสามารถปล่อยรถได้จากถนนที่ index เท่ากับ indexRoadIn ไปยังถนนที่ index เท่ากับ indexRoadOut โดยมีการ return ค่ากลับมาเป็นค่า Car Rate
protected float	findMinLane(int indexRoadIn, int indexRoadout, char Direction, Controller controller) ใช้สำหรับหาค่าความกว้างของถนนที่น้อยที่สุด จากถนนที่ index เท่ากับ indexRoadIn ไปยังถนนที่ index เท่ากับ indexRoadOut
float	findMinSpeed(int indexIn, int indexOut, Controller controller) ใช้สำหรับหาค่าของความเร็วที่น้อยที่สุด (ใช้ค่าน้อยที่สุด ดูได้จากหัวข้อที่ 2.4.4) จากถนนที่ index เท่ากับ indexIn ไปยังถนนที่ index เท่ากับ indexOut
protected float	findRealCar(boolean isRoad1Way, boolean isHeadConnect, int index, char Direction, Controller controller) ใช้สำหรับหาค่าปริมาณรถจริงที่อยู่ในถนนนั้นๆ เรียกใช้จาก useOp() อีกทีหนึ่ง
protected synchronized void	InitControl(int indexOp, Controller controller) ใช้สำหรับการหาค่าที่จำเป็นต่างๆ ในการปล่อยสัญญาณไฟจราจร โดยหาค่าของจังหวะของสัญญาณไฟจราจรที่เป็น indexOp
public void	move(int x, int y) ใช้สำหรับเคลื่อนย้ายวัตถุไปยังตำแหน่งที่ (x, y)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

protected void	moveCar(int indexRoadIn, int indexRoadOut, float amout, char Direction, Controller controller) ใช้สำหรับการเคลื่อนย้ายรถจริงจาก index ของถนนในทางแยกที่ indexRoadIn ไปยังถนนที่ index ของทางแยกเป็น indexRoadOut เป็นจำนวน amout คัน
public void	showAll() ใช้สำหรับแสดง Attribute ต่างๆ ของทางแยกออกทางหน้าจอ (แสดงผลที่ command line บน DOS)
public String	toString() ใช้สำหรับแสดงค่า Attribute ต่างๆ ของทางแยกเป็น String
public synchronized void	upDate() ใช้สำหรับการ update ทางแยก การ update ทางแยกในระบบคือการปล่อยรถให้วิ่งต่อไปที่จังหวะของสัญญาณไฟจราจรเป็น currentStep
protected void	useOp(int indexOp, boolean lastTime, Controller controller) เป็น Method ที่สำคัญมาก ใช้ในการปล่อยรถที่จังหวะของสัญญาณไฟจราจรเป็น indexOp โดยค่า lastTime เป็น flag ในการตรวจสอบว่าเป็นเวลาสุดท้ายหรือไม่ ถ้ามีค่าเป็น true คือเวลาสุดท้ายจะมีการปรับค่าปริมาณรถที่วิ่งได้ให้เป็นจำนวนเต็ม (เหมือนกับของจริงมากที่สุด)

ตารางที่ 3-4 แสดงรายละเอียดของ Method ของคลาสทางแยก

ในการเชื่อมต่อกันของถนนและทางแยก จะมีการเชื่อมต่อกันดังรูปที่ 3-6

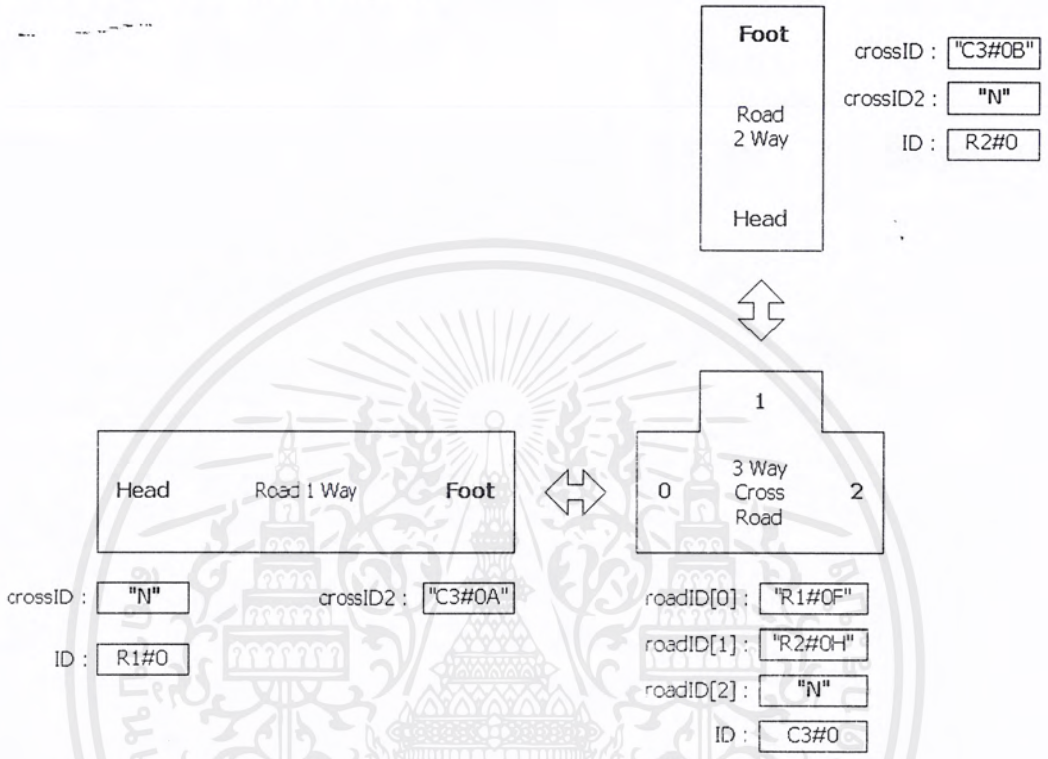
จากรูปที่ 3-6 ด้านบนจะเห็นว่ามีการเชื่อมต่อกันของทางแยกและถนน โดยมีการกำหนดค่า roadID ของทางแยก และมีการกำหนดค่า crossID หรือ crossID2 ของถนนเข้าด้วยกัน โดยมีรูปแบบของการเชื่อมต่อกัน คือ

- ส่วนของทางแยก จะมีการเก็บค่า ID ของถนน ที่ตำแหน่งของอาร์เรย์ตำแหน่งเดียวกับของ index ในทางแยก แล้วตามด้วย "H" หรือ "F" ตามหลัง เช่น R1#0H, R2#12F เป็นต้น
- ส่วนของถนน จะมีการเก็บค่าของ ID ของทางแยกที่ส่วนของ crossID (สำหรับการเชื่อมต่อที่ด้าน Head) หรือที่ crossID2 (สำหรับการเชื่อมต่อที่ด้าน Foot) แล้วตามด้วย "A", "B", "C", "D" หรือ "E" ที่แทนค่าของ index ของถนนในทางแยกโดยเรียงจาก 0, 1, 2, 3, 4 และ 5 ตามลำดับ
- ในทางที่ไม่มีการเชื่อมต่ออยู่จะเก็บเป็นค่า "N" เอาไว้ ซึ่งหมายความว่าเป็นค่าว่าง เวลาจำลองการทำงานของสัญญาณไฟจราจรนั้น สำหรับทางแยกจะมีค่าดังกล่าวไม่ได้ แต่ถ้าเป็นในถนนจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

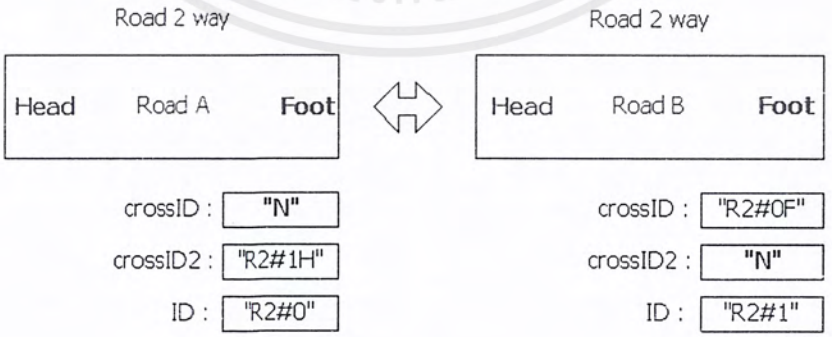
หมายความว่าเป็นการเชื่อมต่อกับปลายเปิด (Open Node) ซึ่งจะมีรถที่วิ่งเข้า หรือวิ่งออกไปได้อย่างอิสระ

หลักของการเก็บการเชื่อมต่อของทางแยกและของถนนนั้นเป็นการเชื่อมต่อกันแบบ Double Link List (ที่ไม่ใช้ Pointer ในการทำงานนั้นมีสาเหตุมาจากว่า ภาษา JAVA เองนั้นไม่มีการใช้ Pointer ในภาษา จึงจำเป็นต้องมีการเชื่อมต่อกันดังกล่าว)



รูปที่ 3-6 แสดงลักษณะของการเชื่อมต่อกันของทางแยก และถนนเข้าด้วยกัน

ส่วนการเชื่อมต่อระหว่างถนน 2 ถนนนั้นก็มีหลักการเช่นเดียวกันกับการเชื่อมต่อของถนนกับทางแยก ดังรูป



รูปที่ 3-7 แสดงการเชื่อมต่อของถนนกับถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อระหว่างถนนกับถนนเองนั้น มีหลักการเชื่อมต่อเช่นเดียวกับการเชื่อมต่อถนนกับทางแยก แต่แทนที่จะเป็นการเก็บ ID ของทางแยกก็เป็นการเก็บ ID ของถนนที่เชื่อมต่อแทนแล้วตามด้วยด้านที่เชื่อมต่อว่าเป็นด้าน Head หรือ Foot โดยจะมีลักษณะของการเชื่อมต่อที่เป็นไปได้ระหว่างถนนกับถนนเป็นดังนี้

- ถนนทางเดียว จะสามารถเชื่อมต่อกันได้ระหว่างถนนทางเดียวกับถนนทางเดียวเท่านั้น และเชื่อมต่อได้ดังนี้

  [Head Foot] ⇔ [Head Foot]

  โดยที่รถจะสามารถวิ่งจากด้าน Head ไปยังด้าน Foot เท่านั้น (สำหรับถนนทางเดียว)
- ถนนสองทาง จะสามารถเชื่อมต่อได้ระหว่างถนนสองทางกับถนนสองทางเท่านั้น และสามารถเชื่อมต่อได้ดังนี้

  [Head Foot] ⇔ [Head Foot]

  [Head Foot] ⇔ [Foot Head]

  [Foot Head] ⇔ [Head Foot]

  [Foot Head] ⇔ [Foot Head]

  จะเห็นได้ว่าสามารถเชื่อมต่อได้ทั้งหมด ทุกแบบ

เราจะสรุปได้ว่า ถนนชนิดใดก็จะเชื่อมต่อได้กับถนนชนิดเดียวกันเท่านั้นไม่สามารถเชื่อมต่างชนิดกันได้

3.2.3.3 คลาส Cross3 (สามแยก)

เป็นคลาสของสามแยก ซึ่งเป็นวัตถุที่ใช้งานจริงในระบบ โดยมีการ Implement Method ของทางแยกเอาไว้ด้วย เช่น display() เป็นต้น รายละเอียดของคลาสสามแยก แสดงได้ในตารางด้านล่างดังนี้

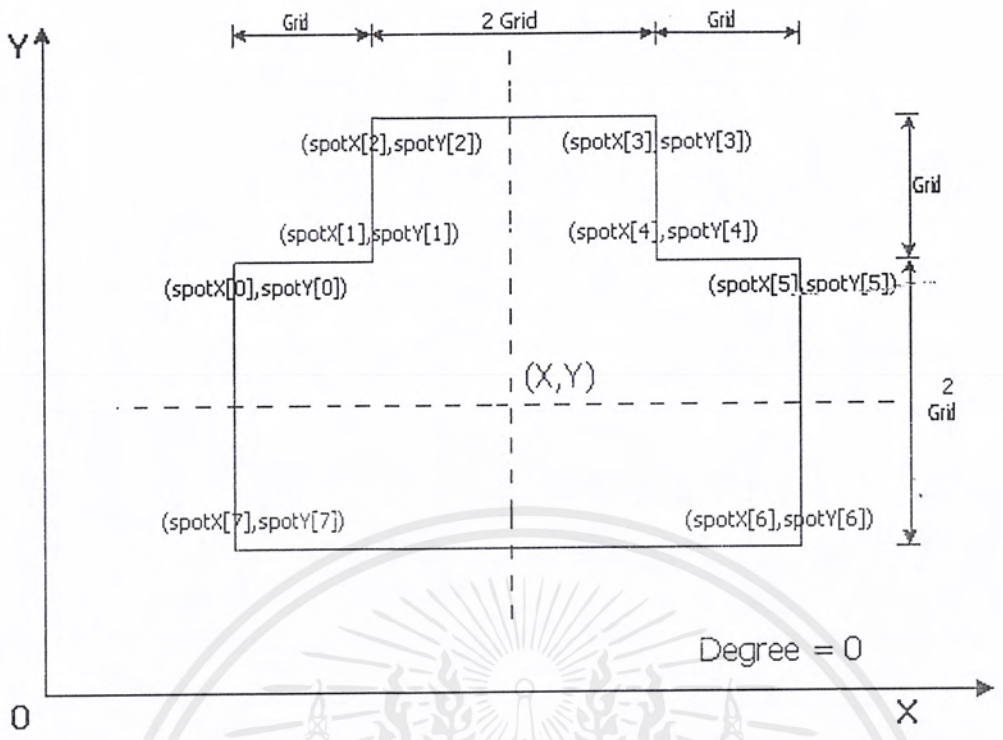
ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
opError[]	String	เป็นค่า op ที่ทำให้การทำงานของสามแยกผิดพลาด จะมีการตรวจสอบ op ของทางแยกก่อนการจำลองการทำงานของโปรแกรมเสมอ โดยที่ opError[] มีค่าเป็น = {“00”, “11”, “22”, “33”, “44”, “03”, “04”, “13”, “14”, “23”, “24”, “30”, “31”, “32”, “34”, “40”, “41”, “42”, “43”}

ตารางที่ 3-5 แสดงรายละเอียดของ Attribute ของคลาสสามแยก

ส่วน Method ของคลาสสามแยกเป็นการ Implement ค่ามาจากคลาสทางแยกทั้งหมด โดยมีจุดประสงค์ของการทำงานของทางแยก

โมเดลของสามแยกที่ได้ออกแบบเอาไว้มีการออกแบบดังรูป

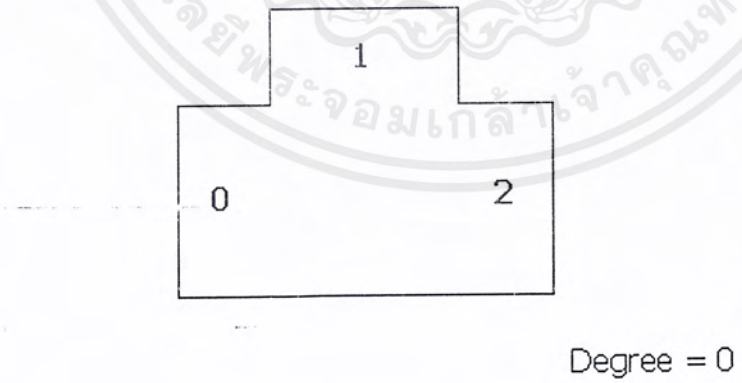
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 แสดงการโมเดลสามแยกในส่วนของการแสดงผลของข้อมูล

การแสดงผลของสามแยกนั้นเป็นการคำนวณค่าจุดต่างๆ ของทางแยก (รวมทั้งการคำนวณจุดของสัญญาณไฟจราจรด้วย) การคำนวณจุดต่างๆ นั้นอาศัยการทำงานของ Computer Graphics (CG) ในการแสดงผลแบบ 2 มิติ ในรูปด้านบนนั้นเป็นการแสดงการแสดงผลของทางแยกที่องศาเท่ากับ 0

การกำหนด index ของถนนที่เข้ามาเชื่อมต่อทางแยกมีการกำหนดเอาไว้ดังรูป



รูปที่ 3-9 แสดง index ของถนนในสามแยก

การกำหนด index ของถนนในทางแยกนั้นเพื่อความง่ายในการอ้างอิง และยังใช้ในส่วนของ op ของทางแยกนั้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.4 คลาส Cross4 (สี่แยก)

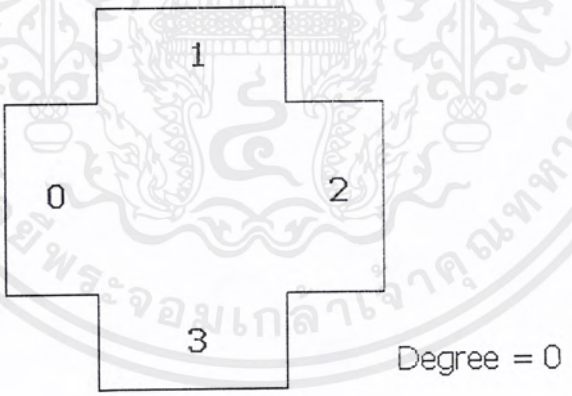
เป็นคลาสของสี่แยก ซึ่งมีการ Implement Method ของทางแยกมาด้วย เช่นเดียวกับคลาส Cross3 โดยจะมีรายละเอียด Attribute ของสี่แยกนั้น แสดงได้ในตาราง

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
opError[]	String	เป็น op ที่เป็นไปไม่ได้สำหรับสี่แยก จะมีการตรวจสอบ op ของสี่แยกที่ Method check() ของทางแยก โดยที่มีค่าเป็นดังนี้ opError[] = {“00”, “11”, “22”, “33”, “44”, “04”, “14”, “24”, “34”, “40”, “41”, “42”, “43”}

ตารางที่ 3-6 แสดงรายละเอียดของ Attribute ของคลาสสี่แยก

การแสดงผลของสี่แยกนั้น แสดงได้ดังรูปที่ 3-11

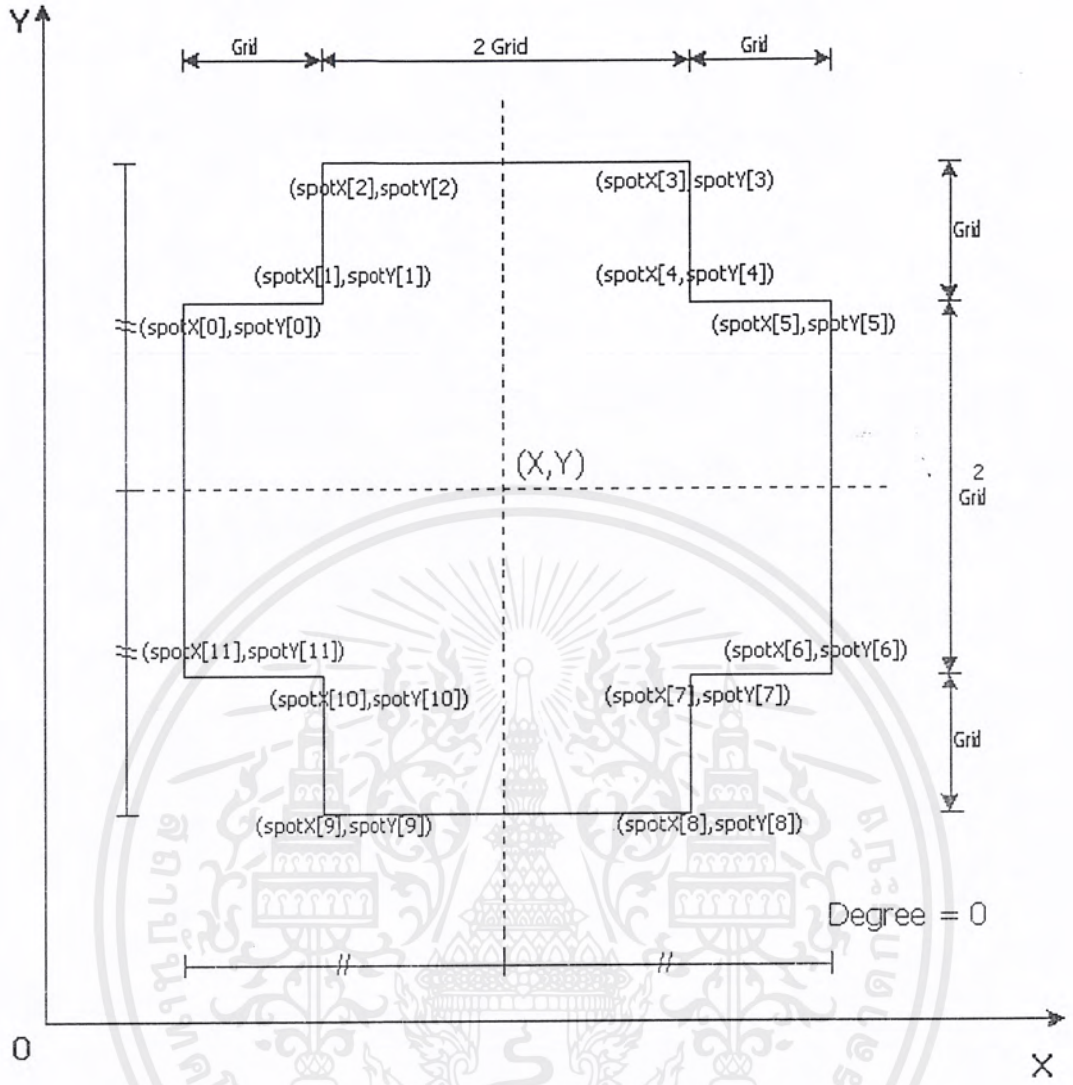
จากรูปที่ 3-11 เป็นการแสดงผลของสี่แยก โดยเป็นการคำนวณจุดต่างๆ ของวัตถุ (มีการคำนวณจุดของวัตถุสัญญาณไฟจราจรด้วย) เช่นเดียวกับสามแยกอาศัยเทคนิคทาง CG ในการหมุนวัตถุ ส่วน index ของถนนในสี่แยกนั้นแสดงได้ดังรูป



รูปที่ 3-10 แสดง index ของถนนในสี่แยก

จากรูปจะเป็นการแสดง index ของถนนในสี่แยก โดย index ที่เห็นในภาพเป็น index ของถนนที่องศาเป็น 0 (ถ้าองศาเป็นอย่างอื่นก็ทำการหมุนวัตถุไป)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-11 แสดงลักษณะโมเดลของสี่แยกในการแสดงผลที่องศาเป็น 0

3.2.3.5 คลาส Cross5 (ห้าแยก)

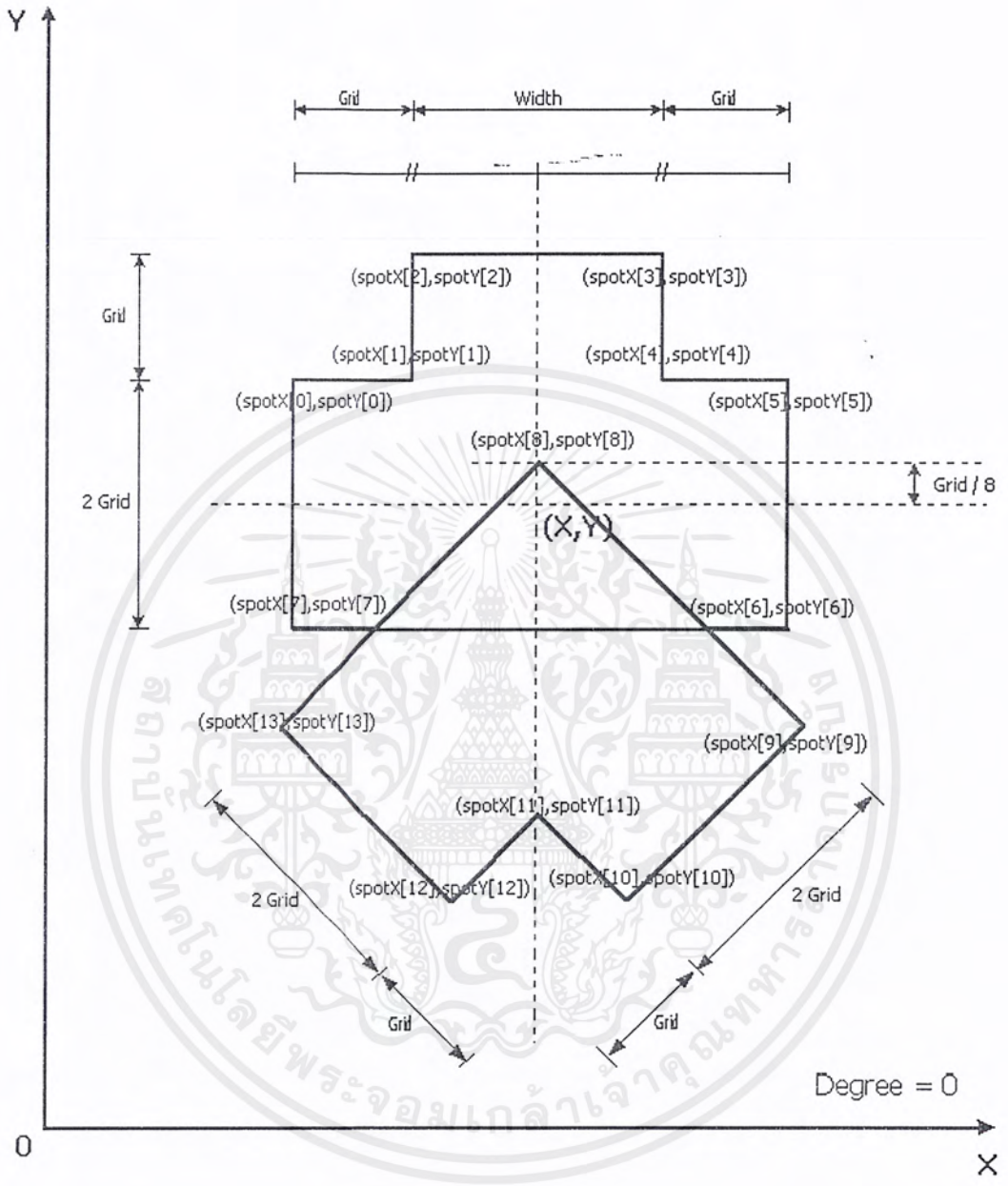
เป็นคลาสของห้าแยกในระบบ สำหรับห้าแยกนั้นพบได้น้อยมาก และมีการเชื่อมต่อของถนนที่ค่อนข้างง่าย (เพราะว่ามีถนนทางเดียวมาเชื่อมต่อด้วย) โดยที่ห้าแยกจะมี Attribute แสดงได้ดังตาราง

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
opError[]	String	Op ที่เป็นไปไม่ได้ (ทำให้การทำงานผิดพลาด) จะตรวจสอบการทำงานของ op ทุกครั้งใน Method check() โดยที่ opError[] จะมีค่าเป็นดังนี้ {“00”, “11”, “22”, “33”, “44”}

ตารางที่ 3-7 แสดงรายละเอียดของ Attribute ของคลาสห้าแยก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

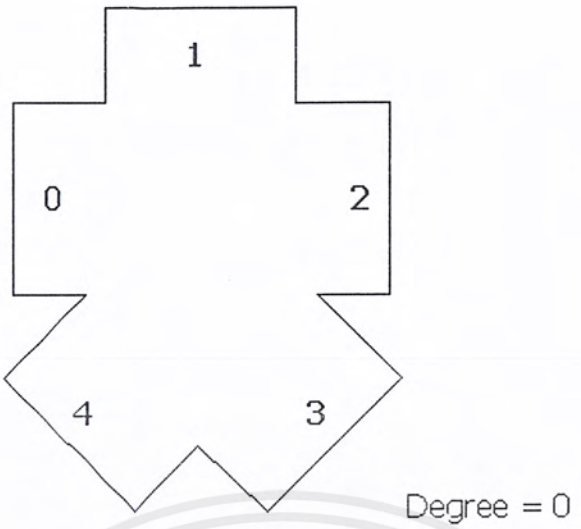
โมเดลของการแสดงผลของห้าแยก แสดงได้ดังรูป



รูปที่ 3-12 แสดงรูปของการแสดงผลของห้าแยกที่องศาเป็น 0

จากรูปด้านบนเป็นการแสดงผลของห้าแยก จะแบ่งการคำนวณของวัตถุออกเป็น 2 ส่วน คือ ส่วนแรกเป็นส่วนของสามแยก และส่วนที่สองเป็นส่วนของหัวเลี้ยวแบบ 90 องศา เมื่อสร้างจุดในหัวเลี้ยวแล้วก็ย้ายจุดมายัง (spotX[8], spotY[8]) และทำการหมุนวัตถุ –45 องศา (หรือ 315 องศานั่นเอง) การคำนวณทั้งหมดนี้จะมีการคำนวณจุดของสัญญาณไฟจราจรด้วย ส่วน index ของถนนในห้าแยกนั้นแสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-13 แสดงลักษณะของ index ของถนนในห้าแยก

ตำแหน่งของ index ของถนนในห้าแยกนั้น จากรูปจะเป็นที่องศาเป็น 0 แต่ถ้าเป็นองศาอื่นก็ทำการหมุนวัตถุไปเอง

3.2.3.6 คลาส Road (ถนน)

เป็นคลาสของถนน เป็นคลาสต้นแบบของถนนทั้งถนนทางเดียวและถนนสองทาง คล้ายๆ กับคลาสทางแยก จะมีบาง Method ที่เป็น abstract Method โดยทำการ Implement ทีหลังในคลาสที่สืบทอดไป จะมีคุณสมบัติของวัตถุเป็นดังนี้

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
AreaL	float	เป็นขนาดถนนที่รถจะวิ่งเลี้ยวซ้าย หาได้จาก AreaL = 2 * leftWidth * speed ; เมื่อ ถนนมี Queue หรือ AreaL = hieght * leftWidth ; เมื่อ ถนนไม่มี Queue ซึ่งมีหน่วยเป็นคัน
AreaR	float	เป็นขนาดถนนที่รถจะวิ่งเลี้ยวขวา หาได้จาก AreaR = 2 * rightWidth * speed ; เมื่อ ถนนมี Queue หรือ AreaR = hieght * rightWidth ; เมื่อ ถนนไม่มี Queue ซึ่งมีหน่วยเป็นคัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AreaS	float	เป็นขนาดถนนที่รถจะวิ่งตรงไป หาได้จาก AreaS = 2 * straightWidth * speed ; เมื่อ ถนนมี Queue หรือ AreaS = hieght * straightWidth ; เมื่อ ถนนไม่มี Queue ซึ่งมีหน่วยเป็นคัน
bL	float	เป็น Buffer ของถนนที่รถจะวิ่งเลี้ยวซ้าย (ดูรูปที่ 3-16) มีหน่วยเป็นคัน
bR	float	เป็น Buffer ของถนนที่รถจะวิ่งเลี้ยวขวา (ดูรูปที่ 3-16) มีหน่วยเป็นคัน
bS	float	เป็น Buffer ของถนนที่รถจะวิ่งตรงไป (ดูรูปที่ 3-16) มีหน่วยเป็นคัน
car[]	float	เป็น Queue ของรถที่อยู่ในถนน มีการสร้าง Queue โดยอาศัย speed ของถนนในการแบ่ง Queue ออกมา ให้ 1 บล็อกมีเวลาในการเคลื่อนที่ของรถจากบล็อกที่ n มายังบล็อกที่ n-1 ใช้ระยะเวลา 1 วินาที และ 1 บล็อกมีขนาดเป็น speed * width (ดูรูปที่ 3-16) มีหน่วยเป็นคัน
color	int	สีของถนน ใช้ในการบอกสถานะของถนนว่ามีการเปลี่ยนแปลงไปเป็นอย่างไร
crossID	String	ส่วนที่ใช้ในการเชื่อมต่อของถนน หรือทางแยกเข้าด้วยกัน (ดูได้รูปที่ 3-6) โดยจะเป็นการเชื่อมต่อที่ด้าน Head ของถนน
crossID2	String	ส่วนที่ใช้ในการเชื่อมต่อของถนน หรือทางแยกเข้าด้วยกัน (ดูได้รูปที่ 3-6) โดยจะเป็นการเชื่อมต่อที่ด้าน Foot ของถนน
degree	int	องศาของถนน (ใช้ในการแสดงผล)
direction	String	ทิศทางของการเคลื่อนที่ของรถจากถนนไปยังวัตถุอื่น จะหาค่าเมื่อตอน InitRoad() ซึ่งจะมีค่ากำหนดให้เป็นดังนี้ "XYZ" เมื่อ X มีค่าเป็น "0" หรือ "1" ถ้ามีค่าเป็น "1" คือรถสามารถเลี้ยวซ้ายได้ แต่ถ้าเป็น "0" คือไม่สามารถเลี้ยวซ้ายได้ Y มีค่าเป็น "0" หรือ "1" ถ้ามีค่าเป็น "1" คือรถสามารถตรงไปได้ แต่ถ้าเป็น "0" คือไม่สามารถตรงไปได้ Z มีค่าเป็น "0" หรือ "1" ถ้ามีค่าเป็น "1" คือรถสามารถเลี้ยวขวาได้ แต่ถ้าเป็น "0" คือรถไม่สามารถเลี้ยวขวาได้
height	int	เป็นค่าความยาวของถนน มีหน่วยเป็น เมตร
ID	String	เป็น Id ของถนน ซึ่งจะมีค่าไม่ซ้ำกันในแต่ละถนน
left	float	เป็นปริมาณรถที่จะเลี้ยวซ้าย มีหน่วยเป็นคัน
leftWidth	float	เป็นความกว้างของเลนที่รถสามารถเลี้ยวซ้ายได้
max	float	เป็นปริมาณรถมากที่สุดที่ถนนสามารถรับได้ มีหน่วยเป็นคัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mode	boolean	เป็นโหมดการคำนวณค่าของรถจาก Queue มา left, right และ straight โดยที่ ถ้ามีค่าเป็น true จะเป็นการสุ่มค่าเปอร์เซ็นต์ของถนนขึ้นมา แต่ถ้ามีค่าเป็น false จะใช้ค่าเปอร์เซ็นต์ที่กำหนดเอาไว้
name	String	ชื่อของถนน
percentLeft	float	เปอร์เซ็นต์การเลี้ยวซ้ายของรถในถนน มีค่าตั้งแต่ 0 - 100
percentRight	float	เปอร์เซ็นต์การเลี้ยวขวาของรถในถนน มีค่าตั้งแต่ 0 – 100
percentStraight	float	เปอร์เซ็นต์การตรงไปของรถในถนน มีค่าตั้งแต่ 0 – 100
QhaveBuild	boolean	เป็น flag ที่ใช้ตรวจสอบว่ามีการสร้าง Queue ในถนนขึ้นหรือไม่ โดยถ้ามีค่าเป็น true แสดงว่ามีการสร้าง Queue แล้ว (แต่ถ้ามีการเปลี่ยนค่าของความยาวของถนน หรือความเร็วของถนนจะต้องมีการสร้าง Queue ใหม่)
QlastMax	float	เป็นขนาดของ Queue สุดท้ายว่ามีขนาดเป็นเท่าใด มีหน่วยเป็นคัน
right	float	เป็นปริมาณรถที่จะเลี้ยวขวาของถนน มีหน่วยเป็นคัน
rightWidth	float	เป็นความกว้างของเลนที่ให้รถเลี้ยวขวา มีหน่วยเป็นเลน
speed	float	เป็นความเร็วเฉลี่ยของรถที่วิ่งได้ในถนน มีหน่วยเป็น คัน/(วินาที*เลน)
spotX[]	int	เป็นจุดที่ใช้แสดงผลของถนนในแกน X
spotY[]	int	เป็นจุดที่ใช้แสดงผลของถนนในแกน Y
straight	float	เป็นปริมาณรถที่จะวิ่งตรงไป มีหน่วยเป็นคัน
straightWidth	float	เป็นความกว้างของเลนที่จะให้รถวิ่งตรงไปของถนน มีหน่วยเป็นเลน
tmpCar	float	เป็นค่าของ Buffer ของถนน มีหน่วยเป็นคัน (ดูรูปที่ 3-16)
tmpCarMax	float	เป็นขนาดของ Buffer ของถนน มีหน่วยเป็นคัน (ดูรูปที่ 3-16)
width	int	เป็นค่าความกว้างของทางแยกในถนน โดยจะได้ว่า width $\geq$ leftWidth + straightWidth + rightWidth มีหน่วยเป็นเลน
X	int	ตำแหน่งแกน X ของถนนในแผนที่ (ใช้ในการแสดงผล)
xMax	int	ค่า X มากที่สุดที่วัตถุมี (ใช้ในการแสดงผล)
xMin	int	ค่า X น้อยที่สุดที่วัตถุมี (ใช้ในการแสดงผล)
Y	int	ตำแหน่งแกน Y ของถนนในแผนที่ (ใช้ในการแสดงผล)
yMax	int	ค่า Y มากที่สุดที่วัตถุมี (ใช้ในการแสดงผล)
yMin	int	ค่า Y น้อยที่สุดที่วัตถุมี (ใช้ในการแสดงผล)

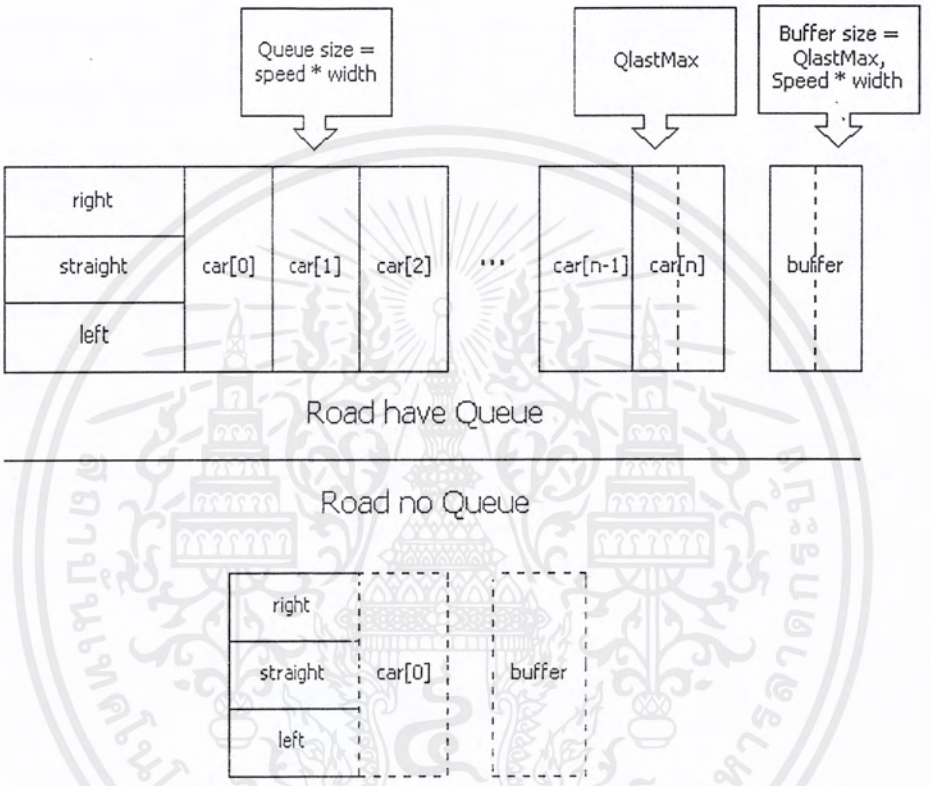
ตารางที่ 3-8 แสดงรายละเอียด Attribute ของคลาสถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากด้านบนเราจะเห็นได้ว่าถนนนั้นสามารถมี Queue หรือไม่มี Queue ก็ได้ โดยถนนที่มี Queue นั้นคือถนนที่มีความยาวมาก ส่วนถนนที่ไม่มี Queue นั้นคือถนนที่สั้นมาก โดยเราจะตรวจสอบได้ว่าถนนนั้นมี Queue หรือไม่จากสมการดังนี้

Block = (Height / (Speed * carHeight))-2

ถ้าค่าของ Block มีค่ามากกว่า 0 แสดงว่าถนนนั้นมี Queue เป็นจำนวน Block แต่ถ้าค่า Block มีค่าน้อยกว่าหรือเท่ากับ 0 แสดงว่าถนนนั้นไม่มี Queue แสดงได้ดังรูป



รูปที่ 3-14 แสดงลักษณะของถนนที่มี Queue และไม่มี Queue

Queue ของถนนนั้นจะมีขนาดเป็น speed * width โดยที่ขนาดของ Queue สุดท้ายนั้นจะมีค่าเป็นได้สองค่าคือ QlastMax หรือว่า speed * width (ในกรณีที่ Queue ข้างหน้าว่างถนนสามารถรับค่าของรถได้เต็มที่) ส่วน Buffer ของถนนนั้นจะมีค่าเท่ากับขนาดของ QlastMax ปัจจุบันของถนน โดยจะมีค่าเป็น speed * width หรือ QlastMax ขึ้นอยู่กับตำแหน่งของ Queue ที่ n-1 ว่าว่างแค่ไหน

ส่วน Method ที่มีในถนนนั้นมีการกำหนด Method เป็น abstract Method อยู่ โดยจะมีการ Implement ในส่วนของวัตถุที่สืบทอดไปทีหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อของ Method	รายละเอียด
abstract public void	addCar(float car, Controller controller) ใช้สำหรับเพิ่มค่ารถเข้าสู่ถนน โดยเป็นการเพิ่มค่าจากด้าน Head โดยจะเป็นการเพิ่มค่าเข้าสู่ Buffer ของถนน (tmpCar)
abstract public boolean	addRealCar() ใช้สำหรับเพิ่มค่าของรถจาก Buffer ไปยัง Queue ของถนน (tmpCar → car[]) ปรับค่าของรถในถนนทั้งหมด และหาปริมาณรถที่รับได้ในวินาทีถัดไป (เป็นการ Sampling ค่าของถนน)
protected float	calculate(float maxCar, float percent) ใช้สำหรับคำนวณเปอร์เซ็นต์ของรถในค่า maxCar โดยเรียกใช้จาก detectDirection() อีกทีหนึ่ง
abstract public boolean	changeColor(Controller controller) ใช้สำหรับตรวจสอบปริมาณรถว่าอยู่ในสถานะใด (Normal, Critical หรือ Jam) แล้วทำการเปลี่ยนสีของถนน โดยถ้า return true แสดงว่ามีการเปลี่ยนสี
abstract public boolean	check(Controller controller) ใช้สำหรับตรวจสอบวัตถุว่ามีข้อผิดพลาดหรือไม่ โดยถ้า return true แสดงว่าไม่มีข้อผิดพลาด แต่ถ้า return false แสดงว่ามีค่าผิดพลาดเกิดขึ้น
public void	connectRoad(String idRoad, boolean isHead, boolean anotherRoadisHead, Controller controller) ใช้สำหรับเชื่อมต่อถนนกับถนนเข้าด้วยกัน (ดูการทำงานได้ในหัวข้อที่ 3.2.3.2 คลาสทางแยกในส่วนของการเชื่อมต่อถนนกับถนน)
protected float[]	detectDirection(String direction, float car, float[] percent) ใช้สำหรับการคำนวณค่าปริมาณรถที่จะวิ่งเลี้ยวซ้าย, ตรงไป หรือเลี้ยวขวา โดย percent ที่ใส่จะเป็นดังนี้ {percentLeft, percentStraight, percentRight} และจะ return ค่ากลับมาเป็นดังนี้ {left car, straight car, right car}
public void	DisconnectRoad(boolean isHead, Controller controller) ใช้สำหรับการยกเลิกการเชื่อมต่อของถนนกับถนน (ดูได้ในหัวข้อที่ 3.2.3.2 คลาสทางแยก)
abstract public void	display(Controller controller) ใช้สำหรับการคำนวณจุดต่างๆ ของถนนเพื่อใช้ในการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

abstract public void	displayDirection(int[] xIn, int[] yIn, int h, int w) ใช้สำหรับคำนวณจุดของทิศทางการวิ่งของรถในถนนนั้นๆ จากด้าน Head ไปยังด้าน Foot
abstract public void	getCar(float car, char carDirection, Controller controller) ใช้สำหรับการลดปริมาณรถในถนน ในทิศทางของ carDirection ซึ่งค่าของ carDirection มีค่าได้ดังนี้ {'l', 'L', 's', 'S', 'r', 'R'}
abstract public float	getCarWait(boolean isHeadConnect) ใช้สำหรับหาค่าของรถที่รออยู่ในถนน โดยจะหาค่าปริมาณรถที่รออยู่ในด้านที่สนใจ ถ้า isHeadConnect เป็น false แสดงว่าหาปริมาณรถที่ด้าน Head (Head → Foot) ดูรูปของการเชื่อมต่อของถนนได้ที่ 3-7
abstract public float	getCarWaitDirection(boolean isHeadConnect, char direction) ใช้สำหรับหาค่ารถที่รออยู่ในถนน โดยมีการระบุทิศทางของถนนที่รออยู่ว่าเป็นทิศทางใด โดยมีค่าเป็นดังนี้ direction = {'L', 'S', 'R'} เมื่อ 'L' เป็นปริมาณรถที่รออยู่ในถนนและจะเลี้ยวซ้าย 'S' เป็นปริมาณรถที่รออยู่ในถนนและจะตรงไป 'R' เป็นปริมาณรถที่รออยู่ในถนนและจะเลี้ยวขวา ส่วนค่า isHeadConnect นั้นถ้ามีค่าเป็น false แสดงค่าหาที่ด้าน Head (Head → Foot)
public float	getInitCar() ใช้หาปริมาณรถที่อยู่ในถนน (นับเฉพาะที่อยู่ใน Queue เท่านั้น)
public synchronized float	getSpace() ใช้สำหรับหาพื้นที่ของถนนที่ว่างทั้งหมด มีหน่วยเป็นคัน (จากด้าน Head → Foot)
public synchronized float	getSpacePossible() ใช้สำหรับหาค่าพื้นที่ว่างที่สามารถรับค่าของรถเข้าไปได้ในถนน (จะถูกเรียกใช้ในการ Sampling ค่าจาก addRealCar() และถูกเรียกใช้ภายหลังจากการเคลื่อนที่ของรถในระบบอีกทีหนึ่ง)
abstract public void	InitRoad(Controller controller) ใช้สำหรับกำหนดค่าต่างๆ ของถนนก่อนการจำลองการทำงานของระบบสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public boolean	isFull() ใช้สำหรับตรวจสอบว่าถนนนั้นเต็มหรือไม่ ถ้า return true แสดงว่าถนนนั้นเต็ม (ในด้าน Head → Foot)
public void	movePosition(int x, int y, Controller controller) ใช้สำหรับเคลื่อนย้ายวัตถุไปยังที่ตำแหน่ง (x, y) ของแผนที่
abstract protected void	moveRoad2Road(Controller controller) ใช้สำหรับการเคลื่อนย้ายรถจากถนนไปยังถนน (ในด้าน Head → Foot)
public void	Qbuild(Controller controller) ใช้สำหรับสร้าง Queue ของถนนขึ้นมา โดยจะทำการลบค่าของ Queue เดิมทั้งหมด
public float	Qcapacity() ใช้สำหรับหาความจุทั้งหมดของ Queue ที่สามารถรับได้ (ในด้าน Head → Foot)
public void	QsetUp() ใช้สำหรับสร้างปริมาณรถทั้งหมด และ AreaL, AreaR, AreaS ของถนน (ในด้าน Head → Foot) จะถูกเรียกใช้งานเมื่อมีการ InitRoad()
public float	Qsum() ใช้สำหรับการรวมค่าของรถทั้งหมดที่อยู่ใน Queue ของถนน (ในด้าน Head → Foot)
public String	QtoString() แสดงปริมาณรถที่อยู่ใน Queue ออกทางหน้าจอ (แสดงผลทาง Command Line ใน DOS) ในด้าน Head → Foot
public String	QtoStringSaveFormat() ใช้สำหรับแปลงค่าของ Queue ที่อยู่ในถนน เป็น String ที่ใช้ในการบันทึกข้อมูล (ในด้าน Head → Foot)
protected void	QupDate() ใช้สำหรับการเคลื่อนย้ายรถที่อยู่ใน Queue ทั้งหมด (ในด้าน Head → Foot)
protected float	random(float maxValue) ใช้ในสุ่มค่าเปอร์เซ็นต์การเลี้ยวของรถในถนน (ในกรณีที่เลือก mode เป็น true)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public void	setInitCar(float car, Controller controller) ใช้สำหรับการกำหนดปริมาณรถที่อยู่ใน Queue ของถนนเป็นปริมาณ car (ในด้าน Head → Foot)
public void	setMaxCar(Controller controller) ใช้สำหรับการหาค่าปริมาณรถสูงสุดที่อยู่ในถนน (ในด้าน Head → Foot)
abstract public void	showAll() ใช้สำหรับการแสดงรายละเอียดของวัตถุ โดยจะแสดงรายละเอียดของวัตถุออกทางหน้าจอ (บน Command Line ของ DOS)
public synchronized float	sumCar() ใช้สำหรับการรวมปริมาณรถทั้งหมดในถนน (ในด้าน Head → Foot)
abstract public String	toString() ใช้สำหรับการแสดงรายละเอียดของวัตถุเป็น String
abstract public void	update(Controller controller) ใช้สำหรับการ update ถนนทั้งหมด (ซึ่งจะเรียกการทำงานที่ Method นี้ในโหมดของการจำลองการทำงานของสัญญาณไฟจราจร)

ตารางที่ 3-9 แสดงรายละเอียดของ Method ของคลาสถนน

จากตาราง จะเห็นได้ว่ามีหลาย Method ที่เป็น abstract Method และมีการใช้งานของ Method จริงในวัตถุที่สืบทอดอีกทีหนึ่ง ในคลาสของถนนทางเดียว และถนนสองทาง

3.2.3.7 คลาส Road1 (ถนนทางเดียว)

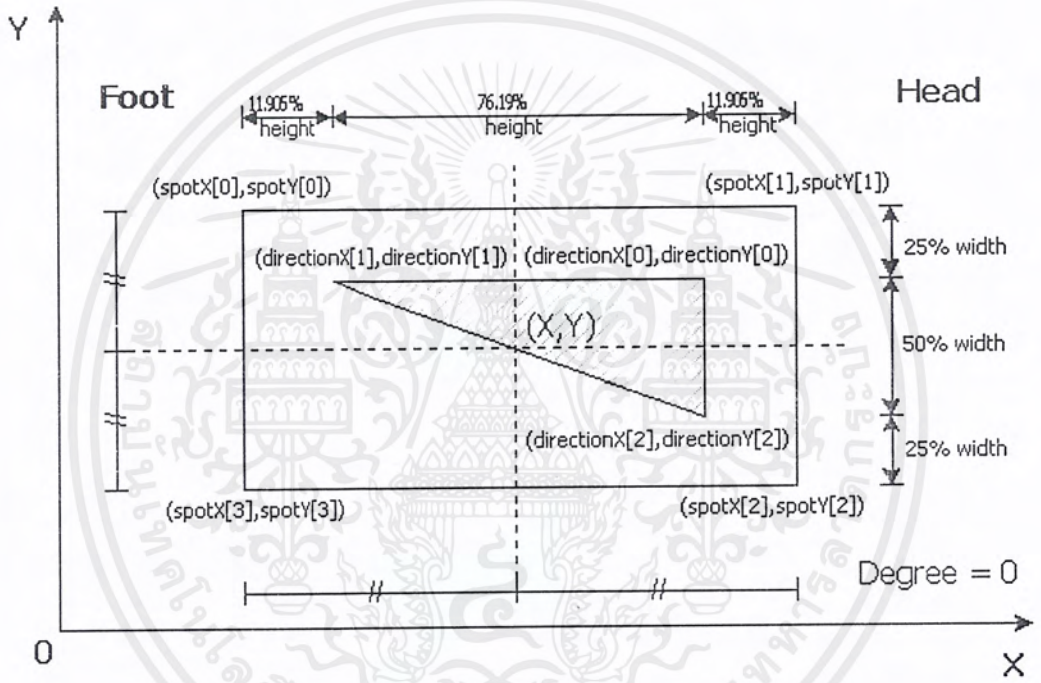
คลาส Road1 เป็นคลาสของถนนทางเดียว หรืออาจจะเรียกว่าถนน 1 Way ก็ได้ โดยคลาสนี้จะสืบทอดมาจากคลาส Road โดยจะมีการ Implement Method ของคลาสต้นแบบเอาไว้ เพื่อให้ใช้งานได้ โดยวัตถุของถนนทางเดียวในระบบจะใช้วัตถุจากคลาสนี้ โดยที่ Attribute ทั้งหมดจะอยู่ในคลาสถนนหมดแล้ว โดยจะมีเพียงแค่บาง Method ที่เพิ่มขึ้นมาเพื่อช่วยในการใช้งานดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อของ Method	รายละเอียด
void	findMinMax() ใช้สำหรับหาค่าสูงสุดและต่ำสุดของจุด โดยจะเก็บค่าที่ได้ลงไปใน xMin, xMax, yMin และ yMax ของวัตถุ

ตารางที่ 3-10 แสดงรายละเอียดของ Attribute ของคลาสถนนทางเดียว

การแสดงผลของถนนทางเดียวจะมีลักษณะดังรูป

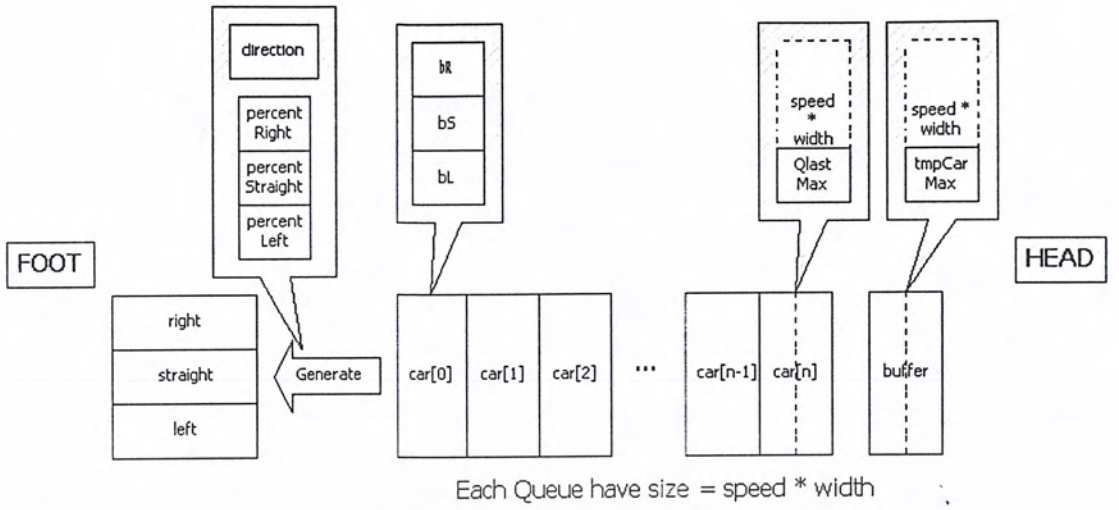


รูปที่ 3-15 แสดงลักษณะของการแสดงผลของถนนทางเดียว

โดยที่สีของถนนเป็น color ซึ่งมีค่าเป็น int ถ้าต้องการจะแสดงผลออกทางหน้าจอ จะเรียกใช้โดยอาศัย Method ของคลาส Controller คือ CompareColor(int color) ในการเซตค่าของสีที่ใช้แสดงผล

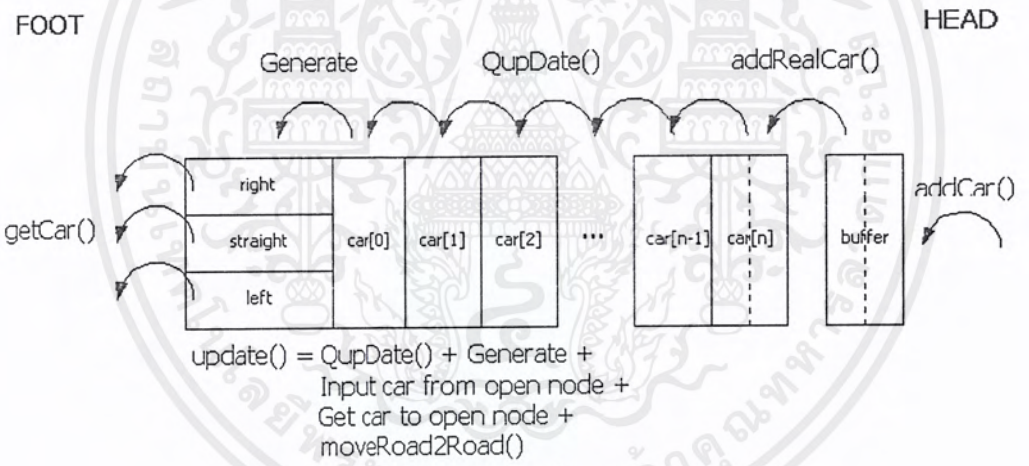
โมเดลของถนนทางเดียวในส่วนของ Data Structure สามารถแสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-16 แสดงโมเดลของถนนทางเดียวในส่วนของ Data Structure

ส่วนโมเดลของ Method ของถนนทางเดียวสามารถแสดงได้ดังรูป



รูปที่ 3-17 แสดงโมเดลของถนนทางเดียวในส่วนของ Method

3.2.3.8 คลาส Road2 (คลาสของถนนสองทาง)

คลาส Road2 เป็นคลาสของถนนสองทาง หรืออาจจะเรียกได้ว่าถนน 2 ways ก็ได้ ซึ่งถนนสองทางนี้จะพบได้โดยทั่วไป มีการสืบทอดมาจากคลาสของถนน และมีการเพิ่ม Attribute ของถนนเข้าไปในส่วนของถนนอีกด้านหนึ่ง และมีการสร้าง Method ที่รองรับการทำงานของส่วนถนนที่เพิ่มขึ้นมาอีกด้วย ซึ่งจะมีลักษณะของวัตถุแสดงได้ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

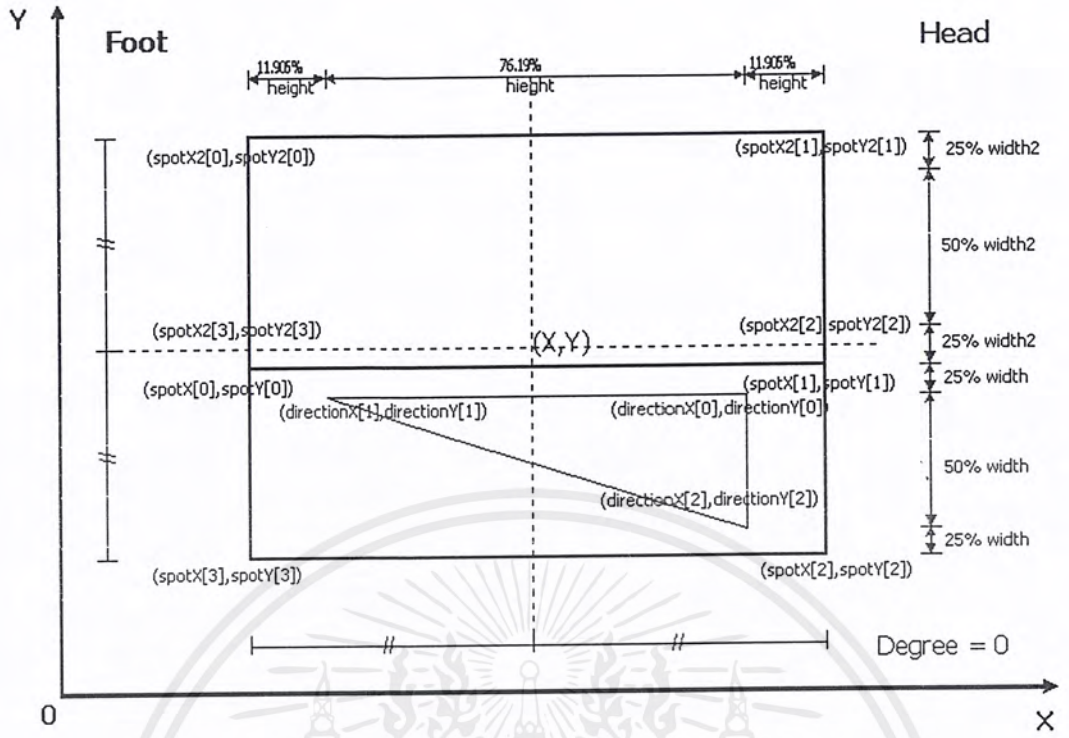
ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
AreaL2	float	เป็นขนาดถนนที่รถจะวิ่งเลี้ยวซ้ายที่ด้าน Foot → Head หาได้จาก AreaL2 = 2 * leftWidth2 * speed2 ; เมื่อ ถนนมี Queue หรือ AreaL2 = hieght2 * leftWidth2 ; เมื่อ ถนนไม่มี Queue ซึ่งมีหน่วยเป็นคัน
AreaR2	float	เป็นขนาดถนนที่รถจะวิ่งเลี้ยวขวาที่ด้าน Foot → Head หาได้จาก AreaR2 = 2 * rightWidth2 * speed2 ; เมื่อ ถนนมี Queue หรือ AreaR2 = hieght2 * rightWidth2 ; เมื่อ ถนนไม่มี Queue ซึ่งมีหน่วยเป็นคัน
AreaS2	float	เป็นขนาดถนนที่รถจะวิ่งตรงไปที่ด้าน Foot → Head หาได้จาก AreaS2 = 2 * straightWidth2 * speed2 ; เมื่อ ถนนมี Queue หรือ AreaS2 = hieght2 * straightWidth2 ; เมื่อ ถนนไม่มี Queue ซึ่งมีหน่วยเป็นคัน
bL2	float	เป็น Buffer ของถนนที่รถจะวิ่งเลี้ยวซ้ายที่ด้าน Foot → Head (ดูรูปที่ 3-19) มีหน่วยเป็นคัน
bR2	float	เป็น Buffer ของถนนที่รถจะวิ่งเลี้ยวขวาที่ด้าน Foot → Head (ดูรูปที่ 3-19) มีหน่วยเป็นคัน
bS2	float	เป็น Buffer ของถนนที่รถจะวิ่งตรงไปที่ด้าน Foot → Head (ดูรูปที่ 3-19) มีหน่วยเป็นคัน
color2	int	สีของถนนที่ด้าน Foot → Head (เปรียบเทียบสีได้จากคลาส Controller)
car2[]	float	Queue ของถนนในด้าน Foot → Head (ดูได้จากรูปที่ 3-19)
direction2	String	ทิศทางการเคลื่อนที่ของรถจาก Queue ไปยังเลี้ยวซ้าย เลี้ยวขวา หรือตรงไป ที่ด้าน Foot → Head
left2	float	ปริมาณรถที่วิ่งเลี้ยวซ้ายของถนนที่ด้าน Foot → Head มีหน่วยเป็นคัน
leftWidth2	float	ความกว้างของเลนของถนนที่ให้รถวิ่งเลี้ยวซ้าย ที่ด้าน Foot → Head
max2	float	ปริมาณรถสูงสุดที่ถนนสามารถรับได้ ที่ด้าน Foot → Head มีหน่วยเป็นคัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

percentLeft2	float	เปอร์เซ็นต์ของรถที่วิ่งเลี้ยวซ้าย ที่ด้าน Foot → Head มีค่าระหว่าง 0 – 100 %
percentRight2	float	เปอร์เซ็นต์ของรถที่วิ่งเลี้ยวขวา ที่ด้าน Foot → Head มีค่าระหว่าง 0 – 100 %
percentStraight2	float	เปอร์เซ็นต์ของรถที่วิ่งตรงไป ที่ด้าน Foot → Head มีค่าระหว่าง 0 – 100 %
QlastMax2	float	ขนาดความกว้างของ Queue สุดท้ายของถนนที่ด้าน Foot → Head มีหน่วยเป็นคัน
right2	float	ปริมาณรถที่วิ่งเลี้ยวขวาของถนนที่ด้าน Foot → Head มีหน่วยเป็นคัน
rightWidth2	float	ความกว้างของเลนของถนนที่ให้รถวิ่งเลี้ยวขวา ที่ด้าน Foot → Head
speed2	float	ความเร็วของรถที่วิ่งได้ในถนนที่ด้าน Foot → Head มีหน่วยเป็นคัน/(วินาที*เลน)
spotX2[]	int	จุดในแกน X ของถนนที่ด้าน Foot → Head (ดูรูปที่ 3-18)
spotY2[]	int	จุดในแกน Y ของถนนที่ด้าน Foot → Head (ดูรูปที่ 3-18)
straight2	float	ปริมาณรถที่วิ่งตรงไปของถนนที่ด้าน Foot → Head มีหน่วยเป็นคัน
straightWidth2	float	ความกว้างของเลนของถนนที่ให้รถวิ่งตรงไป ที่ด้าน Foot → Head
tmpCar2	float	Buffer ของถนนที่ด้าน Foot → Head (ดูได้ที่รูป 3-18) มีหน่วยเป็นคัน
tmpCar2Max	float	ขนาดสูงสุดของ Buffer ที่ถนนสามารถรับได้ ที่ด้าน Foot → Head มีหน่วยเป็นคัน (ดูรูปที่ 3-19)
width2	int	ความกว้างของถนนที่ด้าน Foot → Head มีหน่วยเป็นเลน โดยจะได้ว่า $width2 \geq leftWidth2 + rightWidth2 + straightWidth2$

ตารางที่ 3-11 แสดงรายละเอียดของ Attribute ของคลาสถนนสองทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-18 แสดงลักษณะของการแสดงผลของถนนสองทาง

จากรูปเป็นการแสดงผลของถนนสองทางออกทางหน้าจอ โดยจะเห็นว่ามีค่าของ (spotX, spotY) และ (spotX2, spotY2) ที่มีค่าอยู่ 2 ชุดเพราะว่ามองว่าถนนสองทางนั้นเป็นถนนที่ขยายมาจากถนนทางเดียว จึงมีส่วนของถนนเป็นสองส่วนและในการแสดงผลนั้นก็มีการแสดงผลเป็นของตนเองไม่เกี่ยวกัน

รายละเอียดของ Method ที่เพิ่มขึ้นมาในถนนสองทางนั้นมีดังนี้

ชนิดของ Method	รายละเอียด
public synchronized void	addCar2(float car, Controller controller) ใช้สำหรับการเพิ่มค่ารถเข้าไปในถนนที่ด้าน Foot → Head เป็นปริมาณ car คัน
void	findMinMax() ใช้สำหรับหาค่า X สูงสุดต่ำสุด และค่า Y สูงสุดต่ำสุด เพื่อใช้ในการแสดงผลของถนนสองทาง เรียกใช้จาก display()
public void getCar2	getCar2(float car, char carDirection, Controller controller) ใช้สำหรับลดปริมาณรถที่อยู่ในถนน (เสมือนกับการที่รถวิ่งออกมาจากถนน) ในด้าน Foot → Head (ดูรูปที่ 3-20 แสดงด้านรถวิ่งออกมาจากถนน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

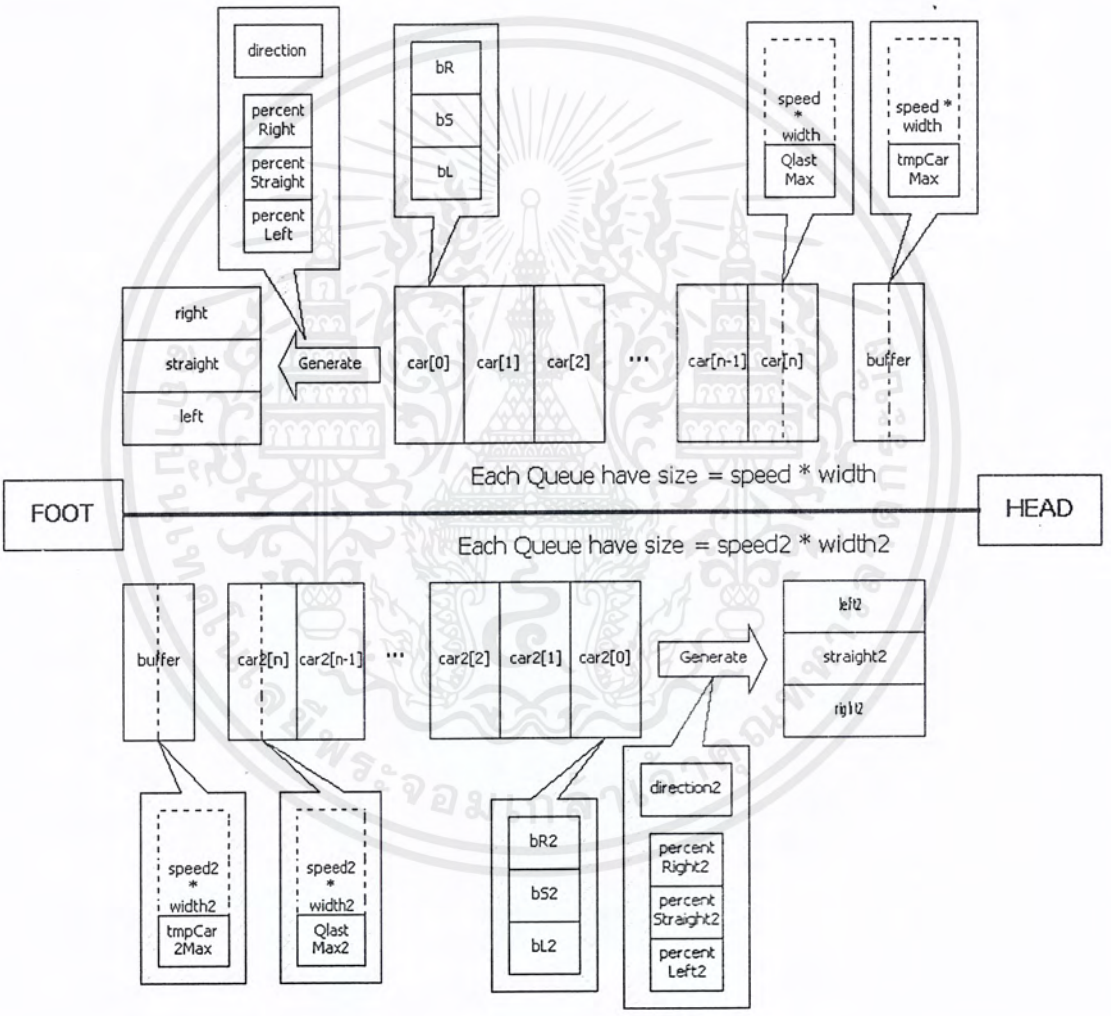
public float	getInitCar2() ใช้สำหรับหาปริมาณรถที่อยู่ในถนน (เฉพาะในส่วนที่อยู่ใน Queue ของถนน) ที่ด้าน Foot → Head มีหน่วยเป็นคัน
public synchronized float	getSpace2() ใช้สำหรับหาพื้นที่ว่างของถนนในด้าน Foot → Head โดยมีหน่วยที่ได้เป็นคัน
public synchronized float	getSpacePossible2() ใช้สำหรับหาปริมาณรถสูงสุดที่ถนนนั้นรับได้ ในเวลา 1 วินาที ที่ด้าน Foot → Head โดยมีหน่วยเป็นคัน
public boolean	isFull2() ใช้ตรวจสอบว่าถนนที่ด้าน Foot → Head นั้นว่างหรือไม่ ถ้า return true แสดงว่าถนนนั้นเต็ม ไม่สามารถรับรถได้อีก
protected void	moveRoad2Road2(Controller controller) ใช้ในการเคลื่อนย้ายรถที่ด้าน Foot → Head ในส่วนของการเชื่อมต่อระหว่างถนนกับถนน เรียกใช้จาก update() อีกทอดหนึ่ง
public float	Qcapacity2() ใช้ในการหาพื้นที่ของ Queue ทั้งหมดของถนนว่ามีเท่าใด ที่ด้าน Foot → Head โดยมีหน่วยเป็นคัน
public void	QsetUp2() ใช้ในการคำนวณค่าเริ่มต้นของ Queue และส่วนของการคำนวณค่าของรถจาก Queue ไปยัง left, right และ straight ที่ด้าน Foot → Head
public float	Qsum2() ใช้ในการรวมจำนวนรถทั้งหมดที่อยู่ใน Queue ของถนนที่ด้าน Foot → Head
public String	QtoStringSaveFormat2() ใช้ในการแปลงค่าของ Queue มาเป็น String เพื่อใช้ในการบันทึกข้อมูลของถนนสองทาง ในส่วนของ Queue ด้าน Foot → Head
public void	setInitCar2(float car, Controller controller) ใช้ในการกำหนดค่าเริ่มต้นของรถที่ด้าน Foot → Head ของถนนเป็นปริมาณ car คัน
public void	setMaxCar2(Controller controller) ใช้ในการคำนวณหาปริมาณรถสูงสุดที่ถนนสามารถรับได้ที่ด้าน Foot → Head

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public float	sumCar2() ใช้ในการรวมปริมาณรถที่อยู่ในถนนทั้งหมดที่ด้าน Foot → Head ของถนน

ตารางที่ 3-12 แสดงรายละเอียดของ Method ของคลาสถนนสองทาง

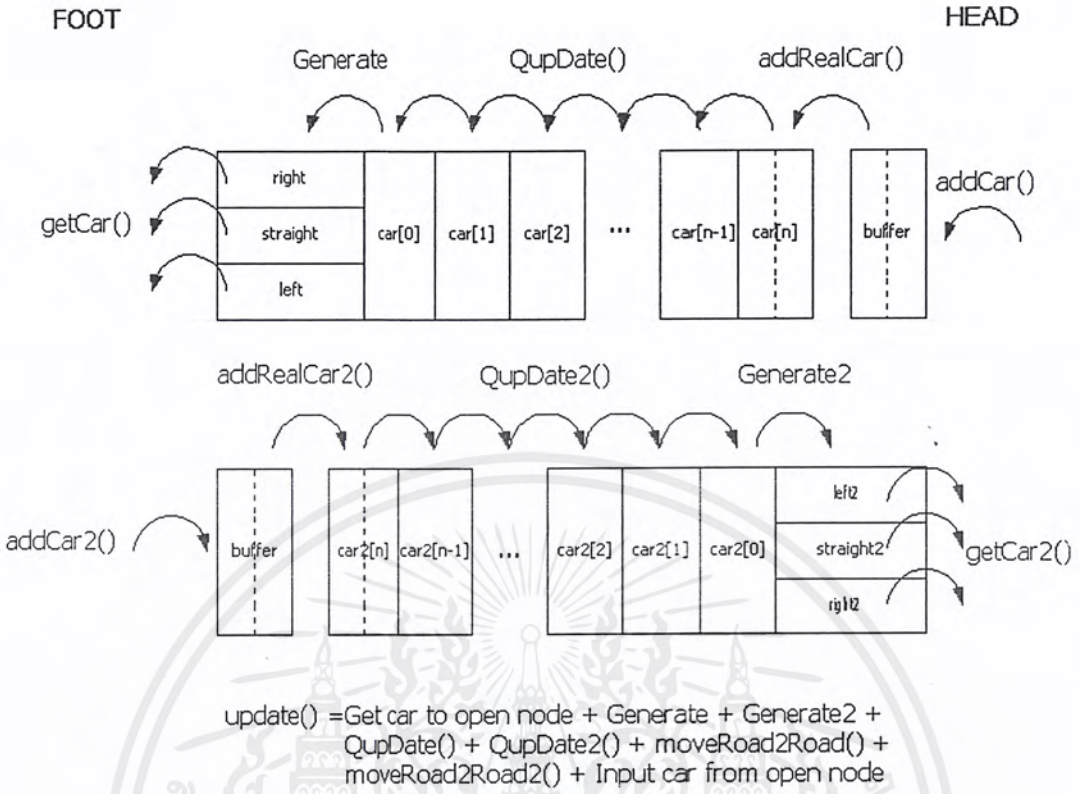
Data Structure ของถนนสองทางนั้นมีส่วนคล้ายกับถนนทางเดียวสองอันมาก แต่ว่ามีการทำงานที่แยกออกเป็นสองส่วน มีลักษณะดังรูป



รูปที่ 3-19 แสดงลักษณะ Data Structure ของถนนสองทาง

ส่วน Method ของถนนสองทางนั้นจะมีชื่อของ Method ที่เหมือนกัน โดยที่ด้าน Head → Foot นั้นจะเป็นชื่อ Method แบบปรกติ ส่วนที่ด้าน Foot → Head นั้นจะมีชื่อเป็นชื่อ Method เดิมในการทำงานเพียงแต่ตามด้วย 2 เท่านั้นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-20 แสดงลักษณะการทำงานของ Method ในถนนสองทาง

3.2.3.9 คลาส TrafficLight (คลาสของสัญญาณไฟจราจร)

เป็นคลาสของสัญญาณไฟจราจร โดยจะใช้ในการแสดงผลของสัญญาณไฟจราจรเท่านั้น ยังไม่มีผลต่อการทำงานของระบบ ซึ่งจะมีรายละเอียดเป็นดังนี้

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
allOp[]	String	เป็น step ของสัญญาณไฟจราจรทั้งหมดที่เป็นไปได้ของทางแยก โดยมีค่าเป็น {"01", "02", "03", "04", "10", "12", "13", "14", "20", "21", "23", "24", "30", "31", "32", "34", "40", "41", "42", "43"}
colorL	Color	เป็นสีของสัญญาณไฟเลี้ยวซ้าย (มีสีอยู่ 3 สีคือ แดง เหลือง เขียว)
colorR	Color	เป็นสีของสัญญาณไฟเลี้ยวขวา (มีสีอยู่ 3 สีคือ แดง เหลือง เขียว)
colorS	Color	เป็นสีของสัญญาณไฟตรงไป (มีสีอยู่ 3 สีคือ แดง เหลือง เขียว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

indexInCrossRoad	char	ตำแหน่งของสัญญาณไฟจราจรที่เทียบเท่ากับ Index ของถนนในทางแยก โดยมีค่าเป็น {'0', '1', '2', '3', '4'}
isYellow	boolean	เป็น Flag แสดงสถานะว่าตอนนี้สัญญาณไฟจราจรแสดงสัญญาณไฟเหลืองอยู่หรือไม่ ถ้ามีค่าเป็น true แสดงว่าแสดงสัญญาณไฟจราจรสีเหลืองอยู่
spotX[]	int	เป็นจุดในแกน X ของวัตถุบนแผนที่
spotY[]	int	เป็นจุดบนแกน Y ของวัตถุบนแผนที่
yellowSignalDelay	int	เป็นระยะเวลาของสัญญาณไฟเหลือง มีค่าตั้งต้นเป็น 5 วินาที

ตารางที่ 3-13 แสดงรายละเอียดของ Attribute ของคลาสสัญญาณไฟจราจร

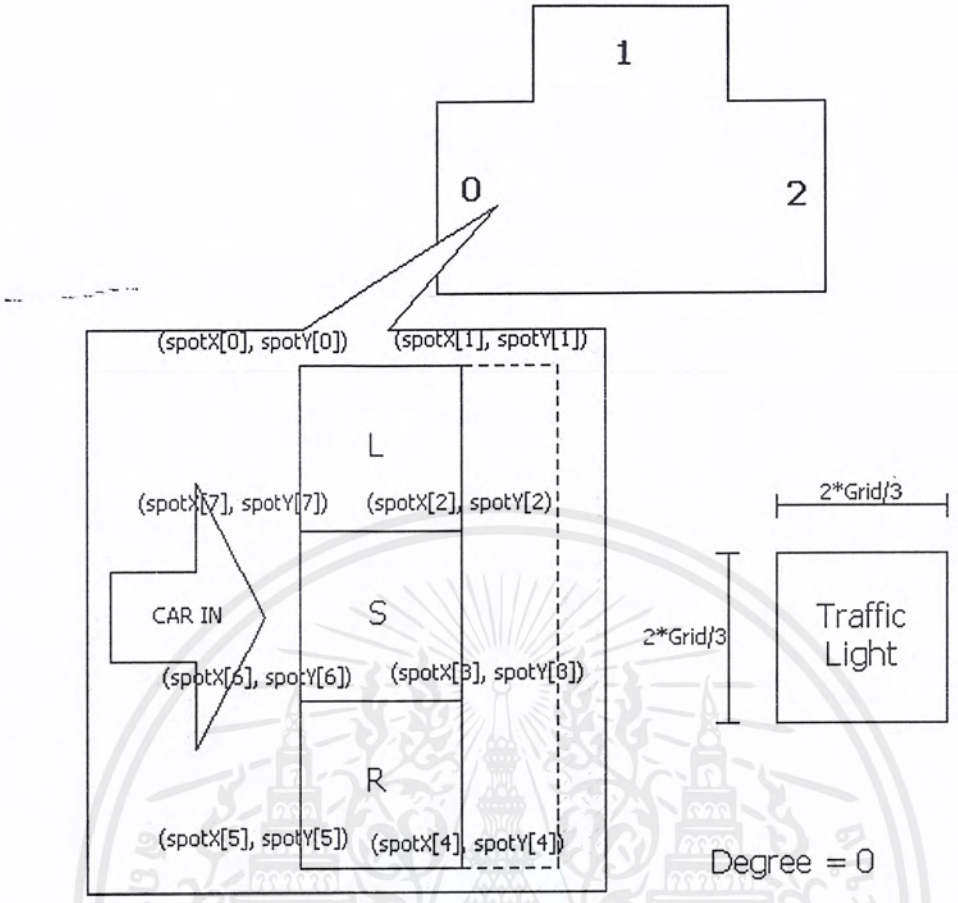
และจะมี Method ที่แสดงได้ดังตาราง

ชนิดของ Method	รายละเอียด
protected char	findSignal(String op, char crossType) ใช้ในการหาสัญญาณไฟจราจรว่าเป็นทิศทางใด โดยจะส่งค่ากลับมาเป็น 'L', 'S' หรือ 'R' เพื่อบอกทิศทางของสัญญาณไฟ
public synchronized boolean	setColor(String op, char crossType, boolean yellowSignal) ใช้ในการเปลี่ยนสีของสัญญาณไฟจราจร โดยเรียกใช้จาก update() ของทางแยกอีกทอดหนึ่ง และมีค่า yellowSignal เพื่อบอกว่าเป็นสัญญาณไฟเหลืองหรือไม่ โดยที่จะ return ค่ากลับมาถ้าเป็น true แสดงว่าสัญญาณไฟมีการเปลี่ยนแปลง

ตารางที่ 3-14 แสดงรายละเอียดของ Method ของคลาสสัญญาณไฟจราจร

วัตถุของสัญญาณไฟจราจรนั้นจะมีลักษณะดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-21 แสดงการแสดงผลของวัตถุสัญญาณไฟจราจร

จากรูปเป็นการแสดงผลของสัญญาณไฟจราจรที่องศาเป็น 0 ถ้ามีการแสดงผลที่องศาอื่นก็ทำการหมุนวัตถุไป โดยที่ขนาดของสัญญาณไฟจราจรจะมีขนาดเป็น (2*grid/3) X (2*grid/3) และถ้าถนนที่เชื่อมต่อกับทางแยกไม่สามารถปล่อยรถเข้าสู่ทางแยกได้ (เป็นถนนทางเดียวที่เชื่อมต่อที่ด้าน Head) ก็จะแสดงผลเป็นสัญญาณไฟสีแดงเสมอ

3.2.3.10 คลาส Index

เป็นคลาสที่ใช้งานทั่วไป จากที่ได้กล่าวมาแล้วว่าภาษา JAVA เองไม่สนับสนุนการทำ Pointer ของโปรแกรม จึงได้มีการออกแบบในส่วนคลาสนี้ขึ้น และใช้ String ในการเชื่อมต่อวัตถุเข้าด้วยกัน โดยอาศัยการทำงานของคลาสนี้ช่วยในการหาตำแหน่งของวัตถุที่ใช้งานจริงในโปรแกรมอีกทีหนึ่ง ซึ่งในการทำงานจริงนั้นจะไม่มีการสร้าง Attribute ของคลาสนี้ แต่จะมีการเรียกใช้ Method ของคลาสนี้เท่านั้น (ใน Method ทั้งหมดของคลาสนี้จะอ้างอิงแบบ Static ทั้งหมด) ซึ่งจะมี Method การทำงานแสดงได้ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของ Method	รายละเอียด
public static String	float2Digit(float value) ใช้สำหรับแปลงค่าจาก float ไปเป็น String ที่มีทศนิยม 2 หลัก เพื่อใช้ในการแสดงผล
public static String	double4Digit(double value) ใช้สำหรับแปลงค่าจาก double ไปเป็น String ที่มีทศนิยม 4 หลัก เพื่อใช้ในการแสดงผล
public static int	indexObject(String object, String name, Controller controller) indexObject(String IDObject, Controller controller) ใช้สำหรับหาค่าของ Index ของวัตถุใน Controller เพื่อนำไปใช้งาน โดยจะใช้งานในส่วนของ Method ที่สองมากกว่า (Method indexObject(String IDObject, Controller controller)
public static int	indexObject2(String object, String ID, Controller controller) ใช้ในการหา index ของวัตถุในคลาส Controller เป็น Method ที่ถูกเรียกใช้งานจาก indexObject() อีกทอดหนึ่ง
public static boolean	isHead(String ID) ใช้ในการตรวจสอบว่ามีการเชื่อมต่อที่ด้าน Head ของวัตถุหรือไม่ (เรียกใช้ในตอนเชื่อมต่อถนนกับทางแยก)
public static boolean	isRoad1Way(String roadID) ใช้ในการตรวจสอบว่า ID นั้นเป็น ID ของถนนทางเดียวหรือไม่
public static String	typeObject(String objectID) ใช้ตรวจสอบชนิดของ ID ว่าเป็นชนิดใด มีค่าที่ส่งกลับไปเป็นดังนี้ {“r1”, “r2”, “c3”, “c4”, “c5”} เป็น ถนนทางเดียว, ถนนสองทาง, สามแยก, สี่แยก และห้าแยกตามลำดับ
public static char	typeObject2(String ID) ใช้ตรวจสอบชนิดของ ID ว่าเป็นชนิดใด มีค่าที่ส่งกลับไปเป็นดังนี้ {‘1’, ‘2’, ‘3’, ‘4’, ‘5’} เป็น ถนนทางเดียว, ถนนสองทาง, สามแยก, สี่แยก และห้าแยกตามลำดับ

ตารางที่ 3-15 แสดงรายละเอียดของ Method ของคลาส Index

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.11 คลาส OpenFile

คลาส OpenFile เป็นคลาสที่ใช้ในการเปิดไฟล์ที่เคยทำการบันทึกเอาไว้มาใช้ โดยถูกเรียกใช้จากคลาส Controller อีกทอดหนึ่ง ซึ่งในคลาสนี้จะไม่มี Attribute ของการทำงาน (เนื่องจากว่าอ้างเป็น Dynamic Variable ทั้งหมดในการทำงาน) มีรายละเอียดของ Method ดังนี้

ชนิดของ Method	รายละเอียด
public static synchronized void	openFile(String fileName, String filePath, Controller controller) throws java.langs.IOException ใช้สำหรับการเปิดไฟล์ที่เคยบันทึกเอาไว้ โดยใส่ค่าเป็นชื่อไฟล์และโฟลเดอร์ของไฟล์นั้นๆ โดยถ้าเกิดข้อผิดพลาดในการเปิดไฟล์จะส่ง Exception ออกมาเป็น IOException
public static void	setAllInitial(Controller controller) ใช้สำหรับกำหนดต่างๆ ของระบบ จะถูกเรียกใช้เมื่อเปิดไฟล์ได้สมบูรณ์แล้ว

ตารางที่ 3-16 แสดงรายละเอียดของ Method ของคลาส OpenFile

3.2.3.12 คลาส SaveFile

เป็นคลาสที่ใช้ในการบันทึกแผนที่ของระบบลงไปเป็นไฟล์ชนิด Text ซึ่งถูกเรียกใช้ผ่านทางคลาส Controller อีกทีหนึ่ง โดยจะมีลำดับของการบันทึกไฟล์เป็นดังนี้

1. ถนนทางเดียว มีลำดับของ Attribute ของถนนเป็นดังนี้

สัญลักษณ์บอกว่าเป็นถนนทางเดียว (“r1”), name, ID, crossID, crossID2, X, Y, height, width, degree, left, right, straight, percentLeft, percentRight, percentStraight, leftWidth, rightWidth, straightWidth, bL, bR, bS, speed, mode, QhaveBuild, ขนาดของ Queue car[] และ car[]

2. ถนนสองทาง มีลำดับของ Attribute ของถนนเป็นดังนี้

สัญลักษณ์บอกว่าเป็นถนนสองทาง (“r2”), name, ID, crossID, crossID2, X, Y, height, width, width2, degree, left, right, straight, left2, right2, straight2, percentLeft, percentRight, percentStraight, percentLeft2, percentRight2, percentStraight2, leftWidth, rightWidth, straightWidth, leftWidth2, rightWidth2, straightWidth2, bL, bR, bS, bL2, bR2, bS2, speed, speed2, mode, QhaveBuild, ขนาดของ Queue car[], car[], ขนาดของ Queue car2[] และ car2[]

3. สามแยก มีลำดับของ Attribute ของถนนเป็นดังนี้

สัญลักษณ์บอกว่าเป็นสามแยก (“c3”), name, ID, X, Y, speed, degree, offsetRoad, roadID[0], roadID[1], roadID[2], ขนาดของ op, op, ขนาดของ time, time, busy, timeOut, currentStep, ขนาดของ car และ car[]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สี่แยก มีลำดับของ Attribute ของถนนเป็นดังนี้

สัญลักษณ์บอกว่าเป็นสี่แยก (“c4”), name, ID, X, Y, speed, degree, offsetRoad, roadID[0], roadID[1], roadID[2], roadID[3], ขนาดของ op, op, ขนาดของ time, time, busy, timeOut, currentStep, ขนาดของ car และ car[]

5. ห้าแยก มีลำดับของ Attribute ของถนนเป็นดังนี้

สัญลักษณ์บอกว่าเป็นห้าแยก (“c5”), name, ID, X, Y, speed, degree, offsetRoad, roadID[0], roadID[1], roadID[2], roadID[3], roadID[4], ขนาดของ op, op, ขนาดของ time, time, busy, timeOut, currentStep, ขนาดของ car และ car[]

6. Controller มีลำดับของ Attribute ของถนนเป็นดังนี้

สัญลักษณ์บอกว่าเป็น Controller (“C”), name, grid, gridWidth, lane, width, height, visionX, visionY, offsetCross3, offsetCross4, offsetCross5, offsetRoad1, offsetRoad2, input, lowCritical, highCritical, zoom, carHeight , idRoad1, idRoad2, idCross3, idCross4, idCross5, edit, normal, critical และ jam

โดยที่จะคั่นแต่ละ Attribute จะใช้เครื่องหมาย “^” เป็นตัวคั่นแต่ละ Attribute เอาไว้ และลงท้ายด้วย “\n” ในการปิดท้ายแต่ละวัตถุ และมีเครื่องหมายบอกว่าเป็นวัตถุชนิดใดอยู่ข้างหน้า (โดยไฟล์ที่ได้จะอยู่ในรูปแบบของ UNIX)

จะมี Method แสดงได้ดังตาราง

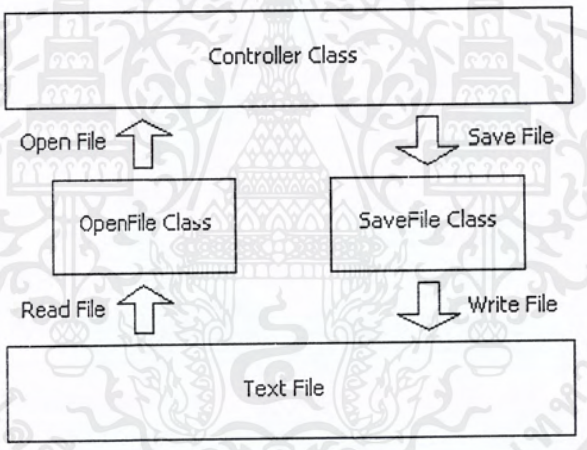
ชนิดของ Method	รายละเอียด
public static void	saveFile(String fileName, String filePath, Controller c) throws java.langs.IOException เป็น Method หลักในการบันทึกไฟล์ ถูกเรียกใช้ผ่านทางคลาส Controller อีกทอดหนึ่ง โดยส่งชื่อของไฟล์ที่ต้องการและโฟลเดอร์ของไฟล์ที่ต้องการ
protected static void	savecontrol(Controller ccc) ใช้ในการบันทึกในส่วนของคลาส Controller ลงไปในไฟล์ ถูกเรียกใช้จาก Method saveFile() อีกทีหนึ่ง
protected static void	savecross3(Controller ccc) ใช้ในการบันทึกในส่วนของวัตถุสามแยกลงไปในไฟล์ ถูกเรียกใช้จาก Method saveFile() อีกทีหนึ่ง
protected static void	savecross4(Controller ccc) ใช้ในการบันทึกในส่วนของวัตถุสี่แยกลงไปในไฟล์ ถูกเรียกใช้จาก Method saveFile() อีกทีหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

protected static void	savecross5(Controller ccc) ใช้ในการบันทึกในส่วนของวัตถุห้าแยกลงไปในไฟล์ ถูกเรียกใช้จาก Method saveFile() อีกทีหนึ่ง
protected static void	saveroad1(Controller ccc) ใช้ในการบันทึกในส่วนของวัตถุถนนทางเดียวลงไปในไฟล์ ถูกเรียกใช้จาก Method saveFile() อีกทีหนึ่ง
protected static void	saveroad2(Controller ccc) ใช้ในการบันทึกในส่วนของถนนสองทางลงไปในไฟล์ ถูกเรียกใช้จาก Method saveFile() อีกทีหนึ่ง

ตารางที่ 3-17 แสดงรายละเอียดของ Method ของคลาส SaveFile ในระบบ

การเปิดไฟล์ และการบันทึกไฟล์จะมีรูปแบบคร่าวๆ ดังรูป



รูปที่ 3-22 แสดงลักษณะของการเปิดไฟล์ และการบันทึกไฟล์อย่างง่ายๆ

3.2.3.13 คลาส Cal

เป็นคลาสที่ใช้ในการคำนวณจุดต่างๆ ภายในระบบทั้งหมด (ใช้ในส่วนของการแสดงผลของระบบ) ซึ่งภายในคลาสนี้จะเป็นการเรียกใช้ Method ต่างๆ ของคลาสเท่านั้น ไม่มีการสร้างวัตถุจากคลาสดังกล่าว มี Method ของคลาสเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

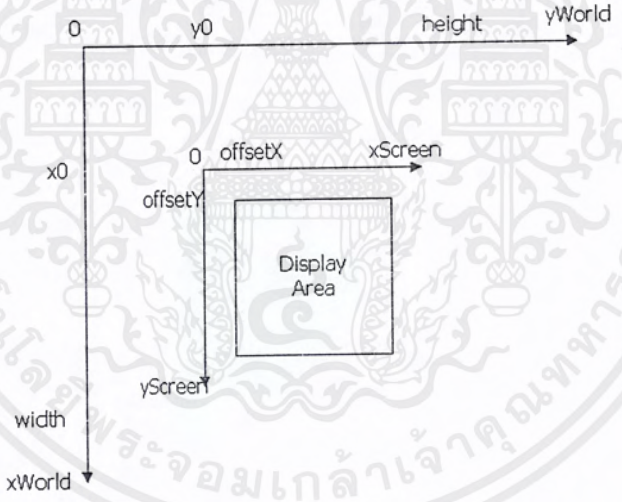
ชนิดของ Method	รายละเอียด
public static int	compare(double value) compare(float value) ใช้ในการแปลงค่าจาก double หรือ float ไปเป็น int
public static double	cos(int degree) ใช้ในการหาค่า cosine ของมุม
public static int	degreeScreen(int degreeWorld) ใช้ในการแปลงค่าองศาของแผนที่ไปเป็นองศาของหน้าจอแสดงผล
public static int	degreeWorld(int degreeScreen) ใช้ในการแปลงค่าจากองศาของหน้าจอไปเป็นองศาในแผนที่
public static int	fact(int value) ใช้ในการหาค่า factorial ของจำนวน
public static int	ObjToInt(Object value) ใช้ในการแปลงค่าของ Object ไปเป็นค่าของ int (ใช้ในกรณีของการแปลงค่าจาก Mouse มาเป็นค่าตัวเลข)
public static double	sin(int degree) ใช้ในการหาค่า sine ของมุม
public static int	xRotate(int x, int y, int degree, int x0, int y0) ใช้ในการหมุนจุด (x, y) รอบจุด (x0, y0) เป็นมุม degree โดยได้ตำแหน่งในแกน x ออกมา
public static int	xScreen(int yWorld, Controller controller) xScreen(float yWorld, Controller controller) xScreen(double yWorld, Controller controller) ใช้ในการแปลงจุดจากจุด y ในแผนที่ไปเป็นจุด x ในหน้าจอ
public static int	xWorld(int yScreen, Controller controller) xWorld(Object yScreen, Controller controller) ใช้ในการแปลงค่าจุดในแกน y ของหน้าจอไปเป็นจุดในแกน x ของแผนที่ (ดูรูปที่ 3-23 สำหรับรายละเอียด)
public static int	yRotate(int x, int y, int degree, int x0, int y0) ใช้ในการหมุนจุด (x, y) รอบจุด (x0, y0) เป็นมุม degree โดยได้ตำแหน่งในแกน y ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public static int	yScreen(int xWorld, Controller controller) yScreen(float xWorld, Controller controler) yScreen(double xWorld, Controller controller) ใช้ในการแปลงค่าของจุด x ในแผนที่ไปเป็นจุด y ในหน้าจอ (ดูรูปที่ 3-23 สำหรับรายละเอียด)
public static int	yWorld(int xScreen, Controller controller) yWorld(Object xScreen, Controller controller) ใช้สำหรับแปลงค่าจุดในแกน x ของหน้าจอไปเป็นจุดในแกน y ของแผนที่ (ดูรูปที่ 3-23 สำหรับรายละเอียด)

ตารางที่ 3-18 แสดงรายละเอียดของ Method ของคลาส Cal

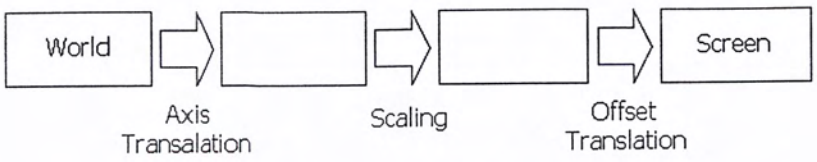
ในโปรแกรมจะมีแกนการทำงานอยู่ 2 แกน คือ แกนของหน้าจอ และแกนของแผนที่ โดยจะได้ความสัมพันธ์กันของแกนทั้ง 2 ดังรูป



รูปที่ 3-23 แสดงความสัมพันธ์ของแกนการทำงานทั้งสองแกน

การคำนวณการแสดงผลจากจุดบนแกน World ไปยังแกน Screen นั้นสามารถทำได้ดังรูปที่ 3-24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-24 แสดงลักษณะของการแปลงค่าจากแกน World ไปยังแกน Screen

จากความสัมพันธ์กันดังกล่าว เราสามารถสร้าง Transformation Matrix ได้ดังนี้

from

$$P' = P \cdot T$$

$$[\, y_s \; x_s \; 1 \,] = [x_w \; y_w \; 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ offset_y & offset_x & 1 \end{bmatrix}$$

$$[\, y_s \; x_s \; 1 \,] = [x_w \; y_w \; 1] \cdot \begin{bmatrix} S & 0 & 1 \\ 0 & S & 1 \\ -x_0S & -y_0S & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ offset_y & offset_x & 1 \end{bmatrix}$$

$$[\, y_s \; x_s \; 1 \,] = [x_w \; y_w \; 1] \cdot \begin{bmatrix} S & 0 & 0 \\ 0 & S & 0 \\ -x_0S+offset_y & -y_0S+offset_x & 1 \end{bmatrix}$$

$$[\, y_s \; x_s \; 1 \,] = [Sx_w-x_0S+offset_y \;\; Sy_w-y_0S+offset_x \;\; 1]$$

$$\therefore \quad y_s = S(x_w-x_0)+offset_y \quad , \quad x_s = S(y_w-y_0)+offset_x$$

$$\therefore \quad x_w = x_0 + (y_s-offset_y)/S \quad , \quad y_w = y_0 + (x_s-offset_x)/S$$

when $x_w = x_{world}$, $y_w = y_{world}$

$x_s = x_{screen}$, $y_s = y_{screen}$

$offset_x$ = offset of screen in x-axis ,

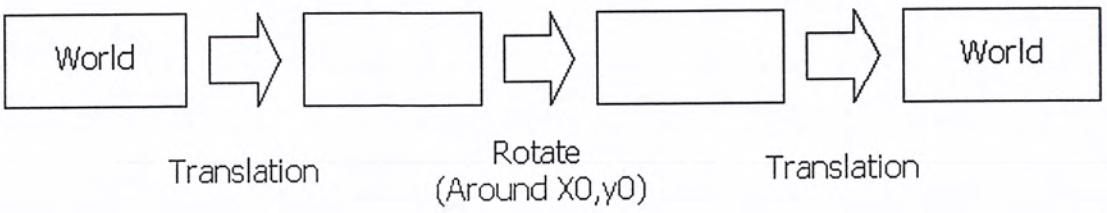
$offset_y$ = offset of screen in y-axis

S = Scaling factor

รูปที่ 3-25 แสดงขั้นตอนการแปลงค่าจากแกนของแผนที่ไปเป็นแกนของหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการหมุนวัตถุที่จุดใดๆ บนแผนที่นั้นเราจะสามารถหมุนได้โดยอาศัยความสัมพันธ์ดังรูป (ซึ่งจะอาศัยเทคนิคการคำนวณของ Computer Graphic ช่วยในการคำนวณ ซึ่งเป็นการทำงานของแผนที่ใน 2 มิติ)



รูปที่ 3-26 แสดงลักษณะของการหมุนจุดที่แกน World

จากรูปด้านบนเราจะสามารถแปลงค่ามาเป็น Matrix ที่ใช้ในการหมุนวัตถุใดๆ ได้ดังนี้

from

$$P' = P \cdot T$$

$$[x'\ y'\ 1] = [x\ y\ 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix}$$

$$[x'\ y'\ 1] = [x\ y\ 1] \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_0\cos\theta+y_0\sin\theta & -x_0\sin\theta-y_0\cos\theta & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix}$$

$$[x'\ y'\ 1] = [x\ y\ 1] \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ x(1-\cos\theta)+y_0\sin\theta & y_0(1-\cos\theta)-x_0\sin\theta & 1 \end{bmatrix}$$

$$[x'\ y'\ 1] = [x\cos\theta-y\sin\theta+x_0(1-\cos\theta)+y_0\sin\theta \quad x\sin\theta+y\cos\theta+y_0(1-\cos\theta)-x_0\sin\theta\ 1]$$

then

$$x' = x\cos\theta - y\sin\theta + x_0(1-\cos\theta)+y_0\sin\theta,$$
$$y' = x\sin\theta + y\cos\theta + y_0(1-\cos\theta)-x_0\sin\theta$$

when

(x_0,y_0) = Point to rotate object around

θ = Degree to move object (+ = counter clock)

รูปที่ 3-27 แสดงขั้นตอนการหาสมการของการหมุนที่จุดใดๆ ในแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.14 คลาส RunThread

เป็นคลาสต้นแบบของการจำลองการทำงานของระบบสัญญาณไฟจราจรทั้งหมด โดยจะมี Attribute ของคลาสเป็นดังนี้

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
canEditStep	boolean	เป็น Flag แสดงว่าสามารถตั้ง step ของสัญญาณไฟจราจรเองได้หรือไม่ ถ้ามีค่าเป็น true แสดงว่าสามารถตั้งค่าได้
canEditTime	boolean	เป็น Flag แสดงว่าสามารถตั้งเวลาของสัญญาณไฟจราจรเองได้หรือไม่ ถ้ามีค่าเป็น true แสดงว่าสามารถตั้งค่าสัญญาณไฟจราจรเองได้
controller	Controller	เป็นระบบที่ต้องการจำลองการทำงานของสัญญาณไฟจราจร
crossType	char	เป็นชนิดของทางแยก โดยถ้ามีค่าเป็น '3' คือสามแยก, '4' คือสี่แยก และ '5' คือห้าแยก จะใช้งานก็ต่อเมื่อสามารถกำหนดสัญญาณไฟจราจรเองได้ (canEditStep เป็นจริง)
delayTimeToChange	int	เป็นเวลาของสัญญาณไฟเหลืองของสัญญาณไฟจราจร โดยจะมีค่ากำหนดเอาไว้เป็น 5 วินาที
exit	boolean	เป็น Flag บอกการทำงานของ Thread ว่ามีการทำงานอยู่หรือไม่ ถ้ามีค่าเป็น true แสดงว่าทำการจำลองระบบสัญญาณไฟจราจรไปเรียบร้อยแล้ว
lastStepCross3[]	int	เป็นจังหวะของสัญญาณไฟจราจรที่ได้เคยปล่อยไปแล้วของสามแยก
lastStepCross4[]	int	เป็นจังหวะของสัญญาณไฟจราจรที่ได้เคยปล่อยไปแล้วของสี่แยก
lastStepCross5[]	int	เป็นจังหวะของสัญญาณไฟจราจรที่ได้เคยปล่อยไปแล้วของห้าแยก
nextTimeToPause	long	เป็นเวลาที่กำหนดให้หยุดการจำลองสัญญาณไฟจราจร ถ้าเวลาขณะนั้น (timer ของ Controller) มีค่าเท่ากับค่านี้
pause	boolean	เป็น Flag ในการหยุดการทำงานของสัญญาณไฟจราจรไว้ชั่วคราว ถ้ามีค่าเป็น true ให้หยุดการจำลองสัญญาณไฟจราจรไว้ชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

step	int	เป็นจังหวะของสัญญาณไฟจราจรที่จะเปลี่ยน ในกรณีที่สามารถกำหนดสัญญาณไฟจราจรเองได้ (canEditStep เป็นจริง)
stop	boolean	เป็น Flag ในการหยุดการทำงานของสัญญาณไฟจราจรทั้งหมด ถ้ามีค่าเป็น true คือหยุดการทำงานของสัญญาณไฟจราจร
TimeToChange	long	เป็นเวลาที่ใช้ในการเปลี่ยนสัญญาณไฟจราจรของทางแยก ในกรณีที่สามารถกำหนดจังหวะสัญญาณไฟจราจรเองได้ (canEditStep เป็นจริง)

ตารางที่ 3-19 แสดงรายละเอียด Attribute ของคลาส RunThread

และจะมีรายละเอียดของ Method ของคลาส RunThread เป็นตาราง

ชนิดของ Method	รายละเอียด
protected void	checkPause() ใช้ตรวจสอบว่ามีการหยุดการจำลองของระบบสัญญาณไฟจราจรอยู่หรือไม่ ถ้าใช่ก็จะหยุดการทำงานที่ Method นี้
protected void	checkStop() ใช้ตรวจสอบว่ามีการสั่งหยุดการทำงานอย่างถาวรหรือไม่ (อาจจะหยุดเพราะหมดเวลาการทำงาน หรือ ผู้ใช้สั่งหยุดการทำงานเองก็ได้)
public Controller	exit() ใช้ Method นี้ในการ update Controller หลังจากที่ได้หยุดการทำงานของระบบแล้วเท่านั้น
public Controller	getController() ใช้เพื่อรับค่าของ Controller ที่ได้จำลองการทำงานของสัญญาณไฟจราจรที่เวลานั้น
public boolean	getExit() ใช้รับค่าของระบบว่ามีการหยุดการทำงานแล้วหรือยัง
public long	getNextTimeToPause() ใช้รับค่าว่าเวลาหยุดครั้งต่อไปของการจำลองระบบสัญญาณไฟจราจรเป็นเมื่อใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

public boolean	getPause() ใช้รับค่าของระบบว่ามีการหยุดการทำงานแบบชั่วคราวแล้วหรือยัง โดยถ้าส่งค่ากลับว่าเป็นจริงแสดงว่าระบบได้หยุดการทำงานแบบชั่วคราวแล้ว
public boolean	getStop() ใช้รับค่าของระบบว่ามีการหยุดการทำงานแบบถาวรแล้วหรือยัง โดยถ้าส่งค่ากลับว่าเป็นจริงแสดงว่าระบบได้หยุดการทำงานแบบถาวรแล้ว
public void	pause() ใช้ในการหยุดการทำงานของระบบโดยทันที (หยุดการทำงานของ Thread นั้นโดยตรง)
public void	play() ใช้ในการจำลองการทำงานของระบบสัญญาณไฟต่อ (โดยเป็นการสั่งงานที่ส่วนของ Thread นั้นโดยตรง)
abstract public void	run() ใช้สำหรับจำลองการทำงานของระบบ โดยที่คลาสลูกจะต้อง Implement Method นี้ทีหลัง เป็น Method หลักในการจำลองระบบสัญญาณไฟจราจร
public synchronized void	setBreak(long time) เป็นการกำหนดจุดหยุดการจำลองระบบสัญญาณไฟจราจร ที่เวลาเป็น time
public void	setController(Controller controller) เป็นการกำหนดระบบที่จะใช้ในการจำลองระบบสัญญาณไฟจราจร
public void	setCross3(Cross3 c3, int index) เป็นการแทนที่สามแยกในระบบที่จำลองการทำงาน (ใช้ในการกำหนดเวลาของการทำงาน โดยผู้ใช้ขณะจำลองการทำงานอยู่ canEditTime เป็นจริง)
public void	setCross4(Cross4 c4, int index) เป็นการแทนที่สี่แยกในระบบที่จำลองการทำงาน (ใช้ในการกำหนดเวลาของการทำงาน โดยผู้ใช้ขณะจำลองการทำงานอยู่ canEditTime เป็นจริง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

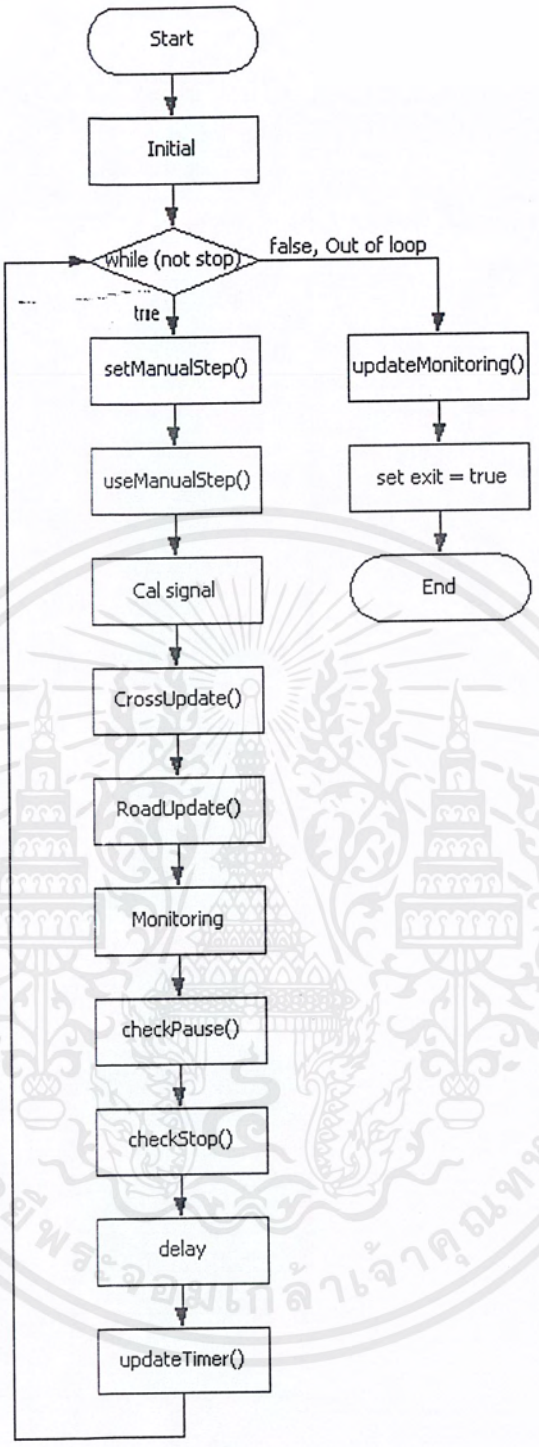
public void	setCross5(Cross5 c5, int index) เป็นการแทนที่ห้าแยกในระบบที่จำลองการทำงาน (ใช้ในการกำหนดเวลาของการทำงาน โดยผู้ใช้ขณะจำลองการทำงานอยู่ canEditTime เป็นจริง)
protected synchronized void	setManualStep() เป็นการสั่งสัญญาณไฟจราจรโดยตรง (สั่งจังหวะของสัญญาณไฟจราจรโดยผู้ใช้ canEditStep เป็นจริง)
public void	setNextTimeToPause(long time) ใช้สำหรับ step การจำลองการทำงานเป็นเวลา time วินาที
public void	setPause(boolean p) เป็นการกำหนดว่าให้ระบบหยุดการจำลองสัญญาณไฟจราจรไว้ชั่วคราวหรือไม่ ถ้า p เป็น true แสดงว่าหยุดการจำลองระบบสัญญาณไฟจราจรไว้ชั่วคราว
public void	start() ใช้ในการเริ่มต้นการจำลองระบบสัญญาณไฟจราจร (เป็นการเริ่มต้นการทำงานของ Thread นั้นโดยตรง)
protected void	updateTimer() ใช้ในการ update เวลาของการจำลองระบบสัญญาณไฟจราจรขึ้นมา 1 วินาที
protected synchronized void	useManualStep() ใช้ในการนำค่าสัญญาณไฟจราจรที่ผู้ใช้ได้เลือกเอาไว้ มาใช้ในทางแยกที่กำหนด (เมื่อระบบสามารถกำหนดระบบสัญญาณไฟจราจรเองได้ canEditStep = true)

ตารางที่ 3-20 แสดงรายละเอียดของ Method ของคลาส RunThread

จากตารางด้านบนจะเห็นได้ว่ามี Method ที่ต้องทำการ Implement ภายหลังคือ run() โดยจะเป็นส่วนที่สำคัญที่สุดในการจำลองระบบสัญญาณไฟจราจร โดยที่จะมีส่วนของการคำนวณสัญญาณไฟจราจรและส่วนของการกำหนดค่าต่างๆ เรียงลำดับดังรูปที่ 3-28

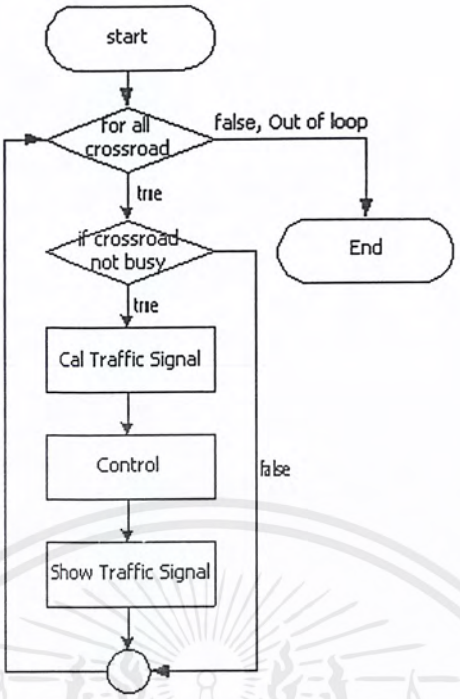
จากรูปจะเห็นได้ว่ามีเฉพาะส่วนของ Cal signal เท่านั้นที่ยังไม่ได้กำหนดขึ้น จะต้องทำการสร้างภายหลัง โดยที่ภายใน cal signal นั้นจะต้องประกอบไปด้วยการทำงานดังรูปที่ 3-29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-28 แสดง Flow Chart ใน Method run() ของคลาส RunThread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-29 แสดง Flow Chart การคำนวณสัญญาณไฟจราจร

3.2.3.15 คลาส RunNormalThread

เป็นคลาสที่จำลองการทำงานของสัญญาณไฟจราจรที่เป็นแบบตั้งเวลา (ตั้งจังหวะของสัญญาณไฟจราจรเอาไว้) เป็นการทำงานของสัญญาณไฟจราจรที่ง่ายที่สุด และได้ทำการสืบทอดมาจากคลาส RunThread อีกทีหนึ่ง

Attribute ของคลาสก็ไม่มีอะไรมาก มีค่าเดียวกับคลาส RunThread เลย ส่วน Method ของคลาสก็เป็น Method ของคลาสต้นแบบ โดยมีส่วนของ Cal Signal เท่านั้นที่มีการทำงานเพิ่มลงไป

3.2.3.16 คลาส RunBestSelectThread

เป็นคลาสการทำงานแบบ AI โดยการคำนวณเพื่อปล่อยรถให้มากที่สุดเท่าที่จะเป็นไปได้ โดยที่มีการสืบทอดมาจากคลาส RunThread รายละเอียดของการคำนวณสัญญาณไฟจราจรจะกล่าวถึงในหัวข้อที่ 3.2.6.1 การคำนวณสัญญาณไฟจราจรแบบเลือกสัญญาณไฟจราจรที่ดีที่สุดต่อไป

รายละเอียดของ Attribute ของคลาส แสดงได้ดังตาราง

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
crossSelect[]	String	เป็น ID ของทางแยกที่เลือกว่ามีการทำงานของสัญญาณไฟจราจรแบบเลือกสัญญาณไฟจราจรที่ดีที่สุด

ตารางที่ 3-21 แสดงรายละเอียดของ Attribute ของคลาส RunBestSelectThread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนรายละเอียดของ Method การทำงานแสดงได้ดังตาราง

ชนิดของ Method	รายละเอียด
protected synchronized void	bestSelectMode(char crossType, int index) เป็นการปล่อยสัญญาณไฟโดยเลือกและคำนวณสัญญาณไฟที่ทำให้รถเคลื่อนที่ได้มากที่สุด (รายละเอียดของการคำนวณสัญญาณไฟที่ทางแยกนั้น แสดงได้ในหัวข้อที่ 3.2.6.2 การคำนวณสัญญาณไฟแบบเลือกสัญญาณไฟที่ดีที่สุด)
protected synchronized boolean	checkSelectCrossRoad(String crossID) ใช้สำหรับตรวจสอบว่าเป็นทางแยกที่ได้เลือกเอาไว้หรือไม่ โดยถ้าส่งค่ากลับมาเป็นจริงแสดงว่าเป็นทางแยกที่ได้เลือกเอาไว้
protected synchronized float	findCarWaitDirection(CrossRoad cr, char Direction, int indexRoad, Controller controller) ใช้สำหรับหาค่ารถที่รออยู่ในถนนในทิศทางของ Direction นั้นๆ ว่ามีการรออยู่เท่าใด (การรอของรถคือ การรอในทิศทางของรถวิ่งตรงไป รถเลี้ยวซ้าย รถเลี้ยวขวา) ซึ่งจะถูกเรียกใช้งานจาก searchOp() อีกทอดหนึ่ง
protected float	findMinValue(float[] value) ใช้สำหรับหาค่าที่น้อยที่สุด จะถูกเรียกใช้งานจาก searchOp() อีกทอดหนึ่ง
protected synchronized float	findSpace(CrossRoad cr, int indexRoad, Controller controller) ใช้สำหรับหาพื้นที่ที่ว่างของถนนว่ามีค่าเป็นเท่าใด จะถูกเรียกใช้งานจาก searchOp() อีกทอดหนึ่ง
protected synchronized void	normalMode(char crossType, int index) ใช้สำหรับการปล่อยสัญญาณไฟที่ทางแยก โดยเป็นการปล่อยสัญญาณไพ่แบบปรกติ (ปล่อยตามเวลาที่ผู้ใช้กำหนดเอาไว้)
protected synchronized float[]	searchOp(String op, CrossRoad cr) ใช้ในการคำนวณหาเวลาของจังหวะของสัญญาณไฟจราจรนั้นๆ ว่าควรจะปล่อยด้วยเวลาเท่าใด ถูกเรียกใช้งานจาก bestSelectMode() อีกทอดหนึ่ง

ตารางที่ 3-22 แสดงรายละเอียดของ Method ของคลาส RunBestSelectThread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.17 คลาส RunBestTimeThread

เป็นการคำนวณสัญญาณไฟจราจรแบบมีการตั้งรูปแบบการเปลี่ยนสัญญาณไฟจราจรที่คงที่ แต่จะมีการคำนวณระยะเวลาของการปล่อยสัญญาณไฟจราจรแต่ละรูปแบบ โดยจะให้เวลาของการทำงานเป็นไปอย่างคุ้มค่าที่สุด (รายละเอียดของการออกแบบการคำนวณสัญญาณไฟจราจรดูได้ในหัวข้อที่ 3.2.6.2 การออกแบบการคำนวณสัญญาณไฟจราจรแบบเลือกเวลาที่ดีที่สุด)

รายละเอียดของ Method และ Attribute ไม่มี เนื่องจากว่าทำการขยายการทำงานออกมาจากคลาส RunBestSelectThread ซึ่งมีการใช้งาน Method เหมือนกัน แต่แตกต่างตรงการเลือกสัญญาณไฟและเวลาของการทำงานเท่านั้น

3.2.3.18 คลาส RunSelectRoadThread

เป็นคลาสที่ใช้ในการคำนวณสัญญาณไฟจราจรเช่นเดียวกับคลาส RunBestSelectThread และคลาส RunBestTimeThread โดยจะเป็นส่วนของการคำนวณสัญญาณไฟจราจรเพื่อให้ถนนที่เลือกนั้นรถติดน้อยที่สุด (ซึ่งผู้ใช้สามารถกำหนดได้ว่าเปอร์เซ็นต์ของถนนที่เลือกนั้นจะสนใจค่าที่เปอร์เซ็นต์เป็นเท่าใด) รายละเอียดของการคำนวณสัญญาณไฟจราจรนั้นสามารถดูได้ที่หัวข้อที่ 3.2.6.3 การคำนวณสัญญาณไฟจราจรแบบเลือกถนนที่ต้องการให้รถติดน้อยที่สุด

รายละเอียดของ Attribute นั้นแสดงได้ดังตาราง

ชื่อของข้อมูล	ชนิดของข้อมูล	รายละเอียด
minpercent	float	เป็นเปอร์เซ็นต์ของรถในถนนที่เลือกว่าสามารถมีรถอยู่ในถนนได้เป็นปริมาณมากที่สุดเป็นเท่าใด (โดยถ้าปริมาณของรถมีค่ามากกว่าค่านี้จะทำการปล่อยรถออกจากถนนดังกล่าวก่อน)
roadname	String	เป็น ID ของถนนที่เลือก

ตารางที่ 3-23 แสดงรายละเอียดของ Attribute ของคลาส RunSelectRoadThread

ส่วนรายละเอียดของ Method ของคลาส RunSelectRoadThread นั้นแสดงได้ดังตาราง

ชนิดของ Method	รายละเอียด
protected boolean	checkop (String optocheck, int numbertocheck) ใช้ในการคำนวณว่าจังหวะของสัญญาณไฟนั้น (optocheck) มีการปล่อยสัญญาณไฟในถนนด้านที่ต่อกับทางแยก (numbertocheck) นั้นหรือไม่ ถ้ามีจะส่งค่า true กลับไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

protected float	findCarWaitDirection (CrossRoad cr, char Direction, int indexRoad, Controller controller) ใช้สำหรับหาปริมาณรถที่รออยู่ในถนนในทิศทางของ Direction ของถนนด้านที่ต่อกับทางแยกที่ด้าน indexRoad
protected String	findconnectside (String crossname) ใช้ในการตรวจสอบว่า ID ของถนนที่รับเข้าต่อกับทางแยกที่ด้านใดของทางแยก
protectd float	findMinValue (float[] value) ใช้ในการหาค่าที่น้อยที่สุด ซึ่งจะถูกเรียกใช้จาก searchOpW() อีกทอดหนึ่ง
protected synchronized float	findSpace(CrossRoad cr, int indexRoad, Controller controller) ใช้ในการหาพื้นที่ว่างของถนน ในถนนด้านที่ต่อกับทางแยกเป็น indexRoad
protected float	percentwaitcarinroad(String idroad, CrossRoad c) รับค่า ID ของถนนกับ ID ของแยกเข้ามาเพื่อคำนวณหาเปอร์เซ็นต์ของจำนวนรถที่ติดรอสัญญาณไฟต่อปริมาณรถมากสุดที่ถนนรับได้ในด้านที่มีทิศทางการเคลื่อนตัวของรถมุ่งหน้าไปที่แยกนั้นๆ
protected int	findnotselectop (CrossRoad c, int selectposition) คำนวณหาจำนวณจังหวะสัญญาณไฟทั้งหมดที่ไม่ได้ปล่อยรถในด้านนั้นๆของทางแยก
protected synchronized float[]	searchOpW (String op, int indexRoadSelect, CrossRoad cr) ใช้ในการคำนวณหาเวลา และปริมาณรถที่ปล่อยได้ของจังหวะของสัญญาณไฟจราจรที่รับเข้ามาโดยสนใจปริมาณรถที่วิ่งในถนนที่ต่อกับทางแยกที่ด้าน indexRoadSelect ซึ่งจะมีรูปแบบของการส่งค่ากลับไปเป็นดังนี้ {เวลาที่ปล่อยได้, ปริมาณรถในถนนที่สนใจที่เคลื่อนที่ได้}

ตารางที่ 3-24 แสดงรายละเอียดของ Method ของคลาส RunSelectRoadThread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 Sequential Diagram

Sequential Diagram เป็นแผนภาพที่ใช้แสดงขั้นตอนการติดต่อระหว่างคลาสต่างๆ ภายในระบบ ซึ่งในโครงงานนี้ได้ทำการออกแบบแผนภาพไว้ 4 แผนภาพดังนี้

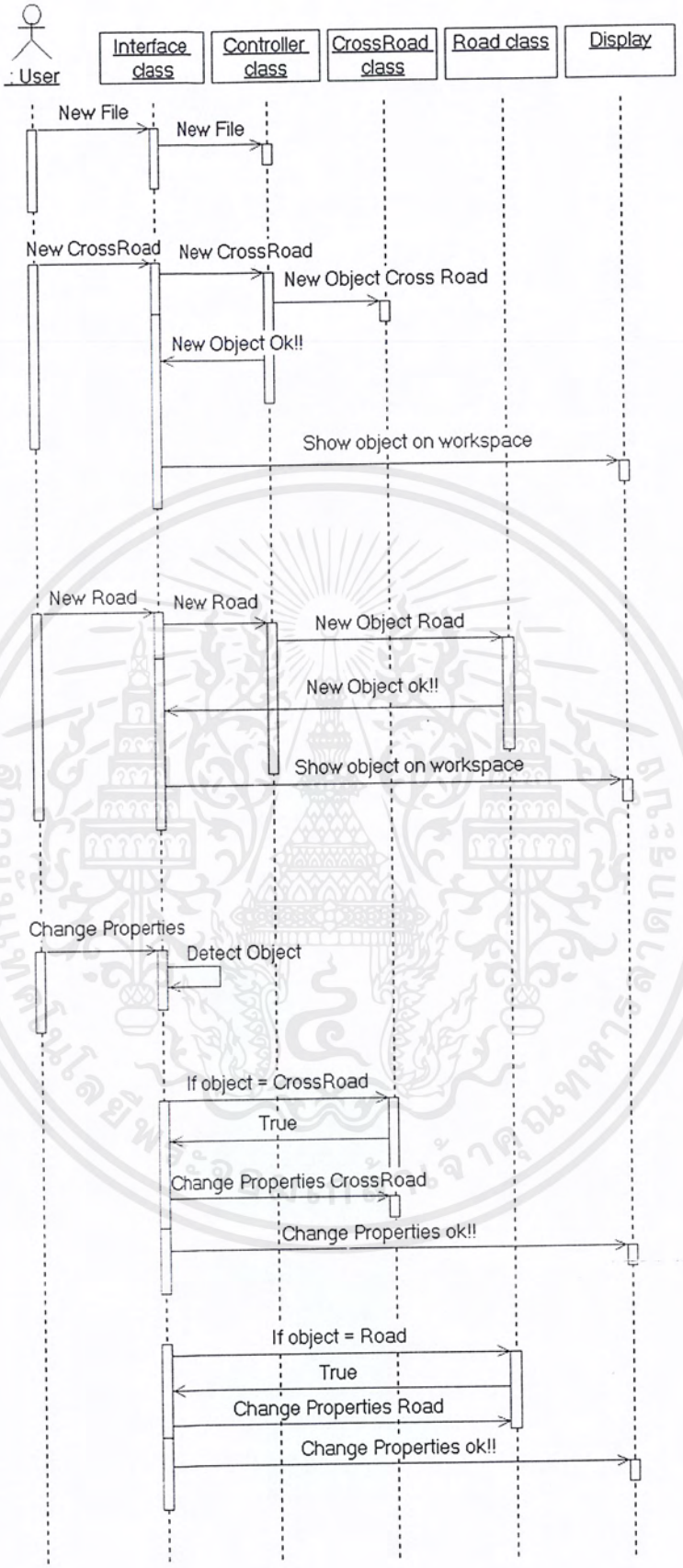
3.2.4.1 Sequential Diagram ของการสร้างและแก้ไขแผนที่

แผนภาพของการสร้าง และแก้ไขแผนที่ มีลักษณะดังรูปที่ 3-30

จากรูป Sequential Diagram ของการสร้างและแก้ไขแผนที่ของระบบมีการทำงานของคลาส ที่เกี่ยวข้องอยู่ 4 ส่วนด้วยกันคือ Interface, Controller, ทางแยก และถนนโดยที่คลาสทางแยกและคลาสถนนเป็นตัวแทนของคลาสลูกของมันเพราะว่า คลาสลูกมีส่วนในการสืบทอด อีกทั้งยังทำหน้าที่เหมือนกันกับคลาสแม่ ส่วนคลาส Interface เป็นส่วนของหน้าจอที่ติดต่อกับผู้ใช้งานในรูปแบบของ Graphic User Interface (GUI) ทั้งโปรแกรมทั้งหมด (รายละเอียดในส่วนของ Interface จะอยู่ในบทที่ 4 การ Implement) ขั้นตอนการทำงานของสร้างและแก้ไขแผนที่มีดังนี้

1. ผู้ใช้ทำการสร้างไฟล์โดยผ่านทางคลาส Interface (อาจจะสร้างโดยโปรแกรมเรียกใช้ตอนต้นหรือผู้ใช้ทำการเรียกใช้เองก็ได้)
2. คลาส Interface ทำการส่ง Message ไปให้คลาส Controller เริ่มทำการกำหนดตัวแปรต่างๆ ที่ใช้ในโปรแกรม
3. ถ้าผู้ใช้ทำการสร้างวัตถุทางแยกขึ้นมา โดยผ่านทางคลาส Interface โดยอาจจะสร้างได้ทั้งสามแยก สี่แยก หรืออาจจะเป็นห้าแยก
4. คลาส Interface ทำการส่ง Message ไปให้คลาส Controller ทำการสร้างวัตถุตามที่ผู้ใช้ เลือกเอาไว้
5. คลาส Controller สร้างวัตถุที่ผู้ใช้เลือก
6. คลาส Controller ส่ง Message บอกคลาส Interface ว่าได้มีการสร้างวัตถุนั้นเรียบร้อยแล้ว
7. ทำการแสดงผลลัพธ์ที่ได้ออกทางหน้าจอ
8. ถ้าผู้ใช้ต้องการสร้างวัตถุถนนขึ้นมา ซึ่งอาจจะเป็นถนนทางเดียว หรือถนนสองทาง โดยผ่านทางคลาส Interface
9. คลาส Interface ส่ง Message บอกคลาส Controller ว่าต้องการสร้างวัตถุถนนขึ้นมาในระบบ
10. คลาส Controller สร้างวัตถุถนนขึ้นมาในระบบ
11. เมื่อวัตถุถนนสร้างเสร็จแล้ว ก็ทำการส่ง Message บอกคลาส Interface ว่ามีการสร้างวัตถุถนนเสร็จแล้ว
12. คลาส Interface แสดงผลลัพธ์ของถนนที่ได้ออกทางหน้าจอ
13. ถ้าเกิดผู้ใช้ต้องการเปลี่ยนแปลงรายละเอียดของถนนหรือทางแยก ก็ทำการสั่งงานผ่านทางคลาส Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-30 แสดง Sequential Diagram ของการสร้างและแก้ไขแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. คลาส Interface ตรวจสอบว่าผู้ใช้ต้องการแก้ไขรายละเอียดของวัตถุชนิดใด ซึ่งอาจจะเป็นถนนทางเดียว, ถนนสองทาง, สามแยก, สี่แยก, ห้าแยก หรือแม้กระทั่งคลาส Controller เองก็ได้ ถ้าเป็นวัตถุของทางแยกเองจะทำการเปลี่ยนแปลงค่าตัวเองทันที
15. ถ้าเป็นวัตถุของทางแยกที่ผู้ใช้ต้องการเปลี่ยนแปลงค่า
16. วัตถุทางแยกตอบรับกลับมาว่าเป็นวัตถุทางแยกที่ผู้ใช้ต้องการแก้ไข
17. คลาส Controller ทำการเปลี่ยนแปลงรายละเอียดของทางแยก
18. คลาส Controller ทำการแสดงผลของทางแยกที่ทำการเปลี่ยนแปลงรายละเอียดแล้วออกทางหน้าจอ
19. หรือถ้าเป็นวัตถุของถนน คลาส Controller ตรวจสอบว่าเป็นวัตถุของถนนหรือไม่
20. คลาสถนนตอบรับกลับมาว่าเป็นวัตถุของถนนที่ผู้ใช้ต้องการเปลี่ยนแปลงค่า
21. คลาส Controller ทำการเปลี่ยนแปลงค่าของถนนนั้น
22. คลาส Controller แสดงผลวัตถุถนนที่ทำการเปลี่ยนแปลงค่าแล้วออกทางหน้าจอ

จากการทำงานด้านบนนั้น ไม่จำเป็นจะต้องมีลำดับของการทำงานเป็นเช่นนั้นตลอดก็ได้ ทั้งนี้นั้นขึ้นอยู่กับผู้ใช้เองว่าต้องการการทำงานเป็นอย่างไรด้วย โดยจะแยกส่วนของการทำงานในแผนภาพออกได้เป็น 3 ส่วนคือ

1. ส่วนของการสร้างวัตถุทางแยก จะใช้ลำดับการทำงานข้อที่ 3. ถึงข้อที่ 7.
2. ส่วนของการสร้างวัตถุถนน จะใช้ลำดับการทำงานข้อที่ 8. ถึงข้อที่ 12.
3. ส่วนของการแก้ไขรายละเอียดของวัตถุ (ไม่ว่าจะเป็นถนน, ทางแยก หรือ Controller เอง) จะใช้ข้อที่ 13. ถึงข้อที่ 22.

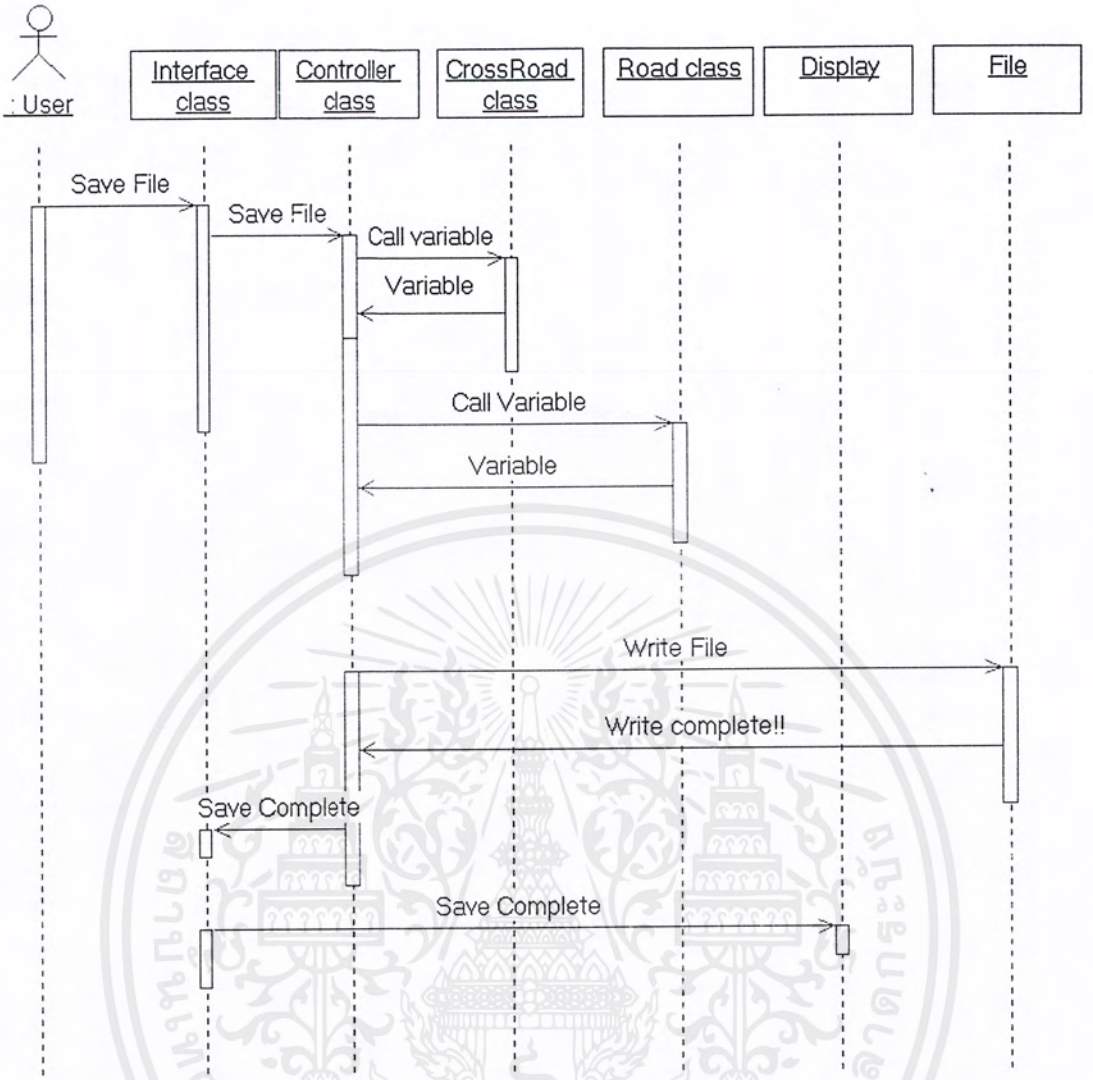
3.2.4.2 Sequential Diagram ของบันทึกไฟล์

Sequential Diagram ของการบันทึกไฟล์จะมีลักษณะดังรูปที่ 3-31

จากรูปที่ 3-31 แสดงลักษณะของ Sequential Diagram ของการบันทึกแผนที่ โดยที่มีการติดต่อกันระหว่างคลาสตามแผนภาพดังนี้

1. ผู้ใช้ทำการสั่งการบันทึกแผนที่ผ่านทางคลาส Interface
2. คลาส Interface ทำการส่ง Message ไปให้คลาส Controller ว่าต้องการบันทึกแผนที่
3. คลาส Controller เรียกรายละเอียดของทางแยกทั้งหมดในระบบ
4. คลาสทางแยกส่งรายละเอียดของทางแยกที่จำเป็นในการบันทึกข้อมูลมาให้คลาส Controller
5. คลาส Controller เรียกรายละเอียดของถนนที่อยู่ในระบบมาทั้งหมด
6. คลาสถนนส่งรายละเอียดของถนนที่จำเป็นมาให้คลาส Controller
7. คลาส Controller เขียนรายละเอียดของระบบลงไปในไฟล์ (รายละเอียดของการทำงานอย่างละเอียดอ่านได้ในหัวข้อที่ 3.2.3.9 คลาส SaveFile)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-31 แสดง Sequential Diagram ของการบันทึกไฟล์

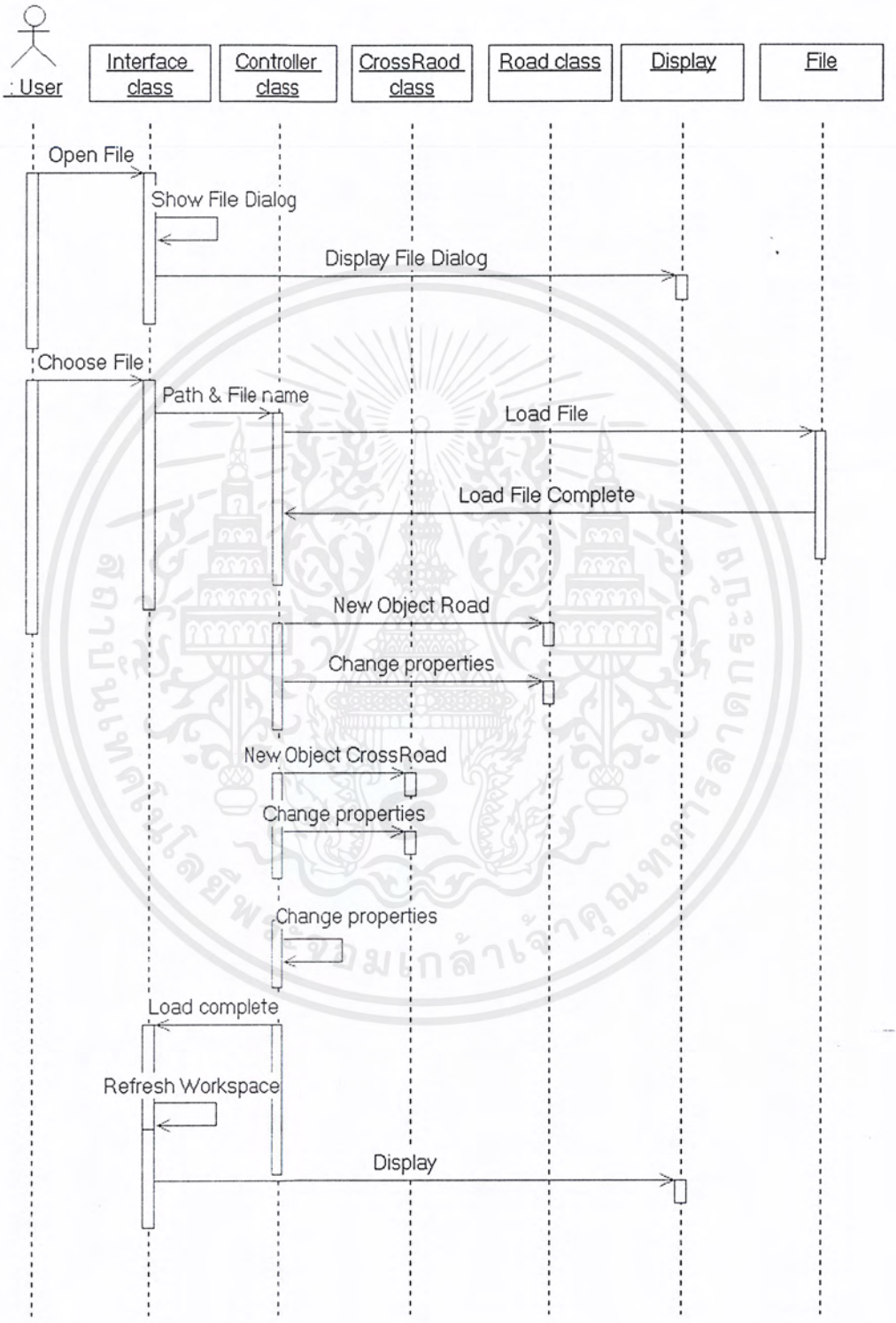
8. เมื่อบันทึกข้อมูลลงไปในไฟล์แล้วทางระบบปฏิบัติการก็จะส่งรายละเอียดว่าบันทึกไฟล์เรียบร้อยแล้ว
9. คลาส Controller ส่ง Message บอกคลาส Interface ว่ามีการบันทึกแผนที่สมบูรณ์แล้ว
10. คลาส Interface แสดงผลว่ามีการบันทึกไฟล์แล้ว

จากการติดต่อกันอย่างคร่าวๆ นั้นคลาส Controller เรียกใช้งานคลาส SaveFile ในการจัดการทั้งหมด โดยที่คลาส Interface เรียกใช้งานผ่านทางคลาส Controller แล้วคลาส Controller เรียกใช้งานผ่านทางคลาส SaveFile อีกทอดหนึ่ง จะไม่มีการเรียกใช้งานคลาส SaveFile โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4.3 Sequential Diagram ของการเปิดไฟล์

เป็นแผนภาพของการทำงานในส่วนของการเปิดไฟล์ที่เคยบันทึกเอาไว้แล้วมาใช้งาน จะมีลักษณะดังรูป



รูปที่ 3-32 แสดงลักษณะของ Sequential Diagram ของการเปิดไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดง Sequential Diagram ของการเปิดไฟล์ซึ่งจะมีขั้นตอนการทำงานเป็นดังนี้

1. ผู้ใช้สั่งการเปิดไฟล์ผ่านทางคลาส Interface
2. คลาส Interface สร้าง Dialog ให้ผู้ใช้เลือกไฟล์ที่ต้องการเปิด
3. แสดง Dialog ของการเลือกไฟล์ออกทางหน้าจอ
4. ผู้ใช้เลือกไฟล์ที่ต้องการจะเปิดบน Dialog ที่โปรแกรมสร้างขึ้น
5. คลาส Interface ส่ง Message บอกคลาส Controller ถึงชื่อไฟล์ที่ต้องการจะเปิด และโฟลเดอร์ที่ไฟล์นั้นอยู่มาให้
6. คลาส Controller ทำการเปิดไฟล์นั้น
7. คลาส Controller ได้รับไฟล์มาแล้ว ในรูปแบบของตัวอักษร (Text File)
8. คลาส Controller สร้างวัตถุของถนนทั้งหมดตามจำนวนบันทึกเอาไว้ในไฟล์
9. คลาส Controller เปลี่ยนแปลงค่าของวัตถุตามไฟล์ที่ระบุเอาไว้
10. คลาส Controller สร้างวัตถุของทางแยกทั้งหมดในระบบตามที่ไฟล์ได้บันทึกเอาไว้
11. คลาส Controller เปลี่ยนแปลงค่าของทางแยกตามไฟล์ที่ระบุเอาไว้
12. คลาส Controller เปลี่ยนแปลงค่าต่างๆ ภายในคลาสตามที่ได้ระบุเอาไว้ในไฟล์
13. คลาส Controller ส่ง Message บอกคลาส Interface ว่ามีการเปิดไฟล์เสร็จเรียบร้อยแล้ว
14. คลาส Interface ทำการแสดงผลรายละเอียดของแผนที่ และค่าต่างๆ ที่เกี่ยวข้องในการทำงาน
15. คลาส Interface ทำการแสดงผลแผนที่ที่ได้ทำการเปิดแล้วออกทางหน้าจอ

การทำงานของคลาส Controller ในส่วนของการเปิดไฟล์นั้น คลาส Controller จะมีการเรียกใช้งานผ่านทางคลาส OpenFile ซึ่งเป็นส่วนในการเปิดไฟล์จริง (ดูรายละเอียดได้ที่หัวข้อ 3.2.3.8 OpenFile) อย่างไรก็ตาม การทำงานจริงๆ นั้นก็มีการสั่งงานผ่านทางคลาส Controller อยู่ดีคลาส OpenFile เป็นเพียงส่วนในการทำงานเท่านั้น

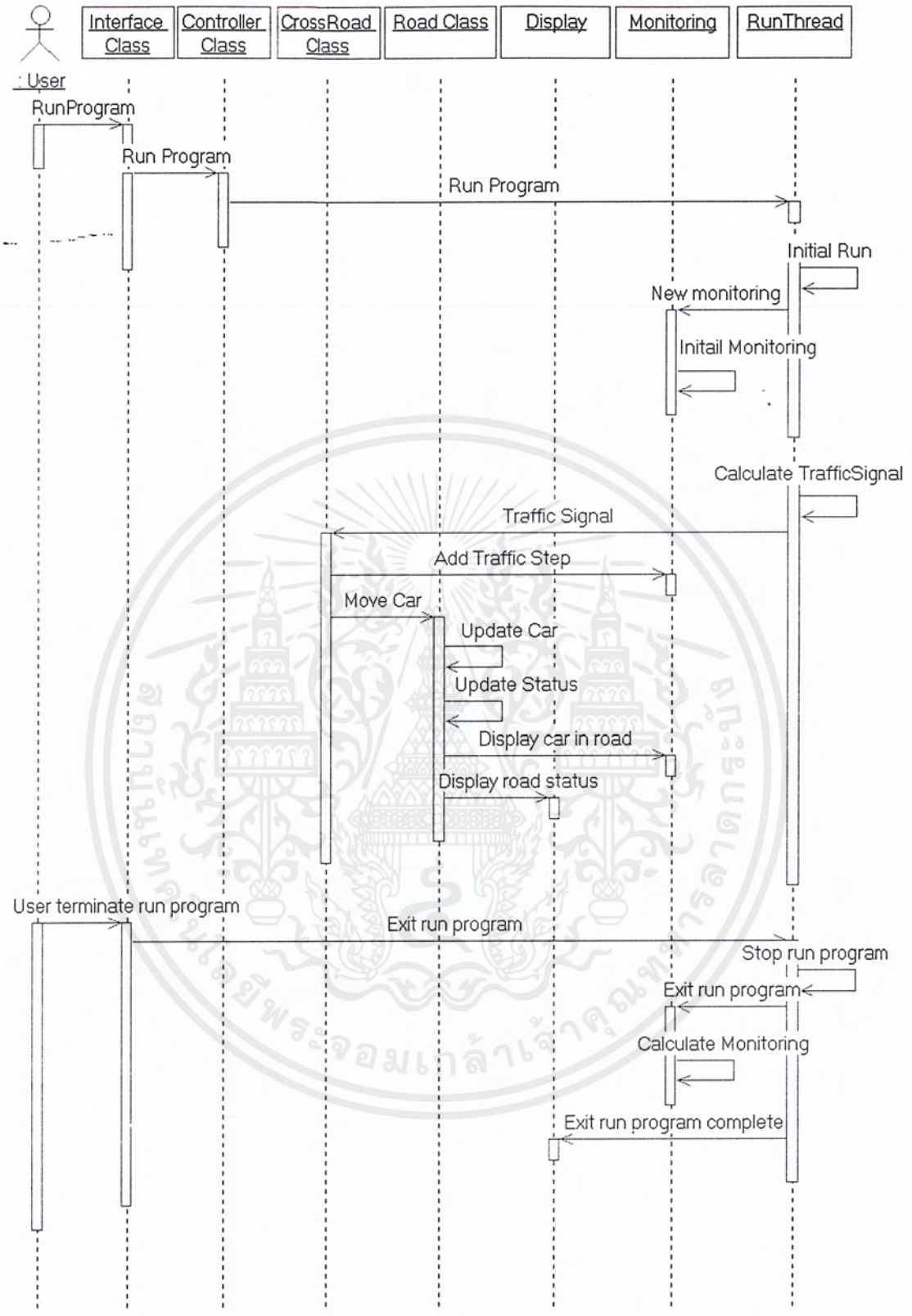
3.2.4.4 Sequential Diagram ของการจำลองการทำงานของสัญญาณไฟจราจร

แผนภาพของการจำลองการทำงานของสัญญาณไฟจราจร (การ Run Program) จะมีลักษณะดังรูปที่ 3-33

จากรูปที่ 3-33 แสดงลักษณะของ Sequential Diagram ของการจำลองการทำงานของระบบสัญญาณไฟจราจร โดยจะเป็นการสร้าง thread ในการจำลองสัญญาณไฟจราจร และในส่วนของการแสดงผลของหน้าจอ ซึ่งในแผนภาพจะเป็นการทำงานในส่วนของการคำนวณสัญญาณไฟจราจรของระบบเท่านั้น

1. ผู้ใช้ทำการสั่งการจำลองการทำงานของสัญญาณไฟจราจร (ไม่ว่าจะเป็น AI mode หรือ Normal mode ก็เหมือนกัน แตกต่างกันที่ Method ที่ทำการเรียกใช้ใน คลาส Controller เท่านั้น) ผ่านทางคลาส Interface
2. คลาส Interface ทำการส่ง Message ไปให้คลาส Controller เลือก mode ที่ใช้ในคลาส Controller ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-33 แสดงลักษณะของ Sequential Diagram ของการจำลองการทำงานของสัญญาณไฟจราจร

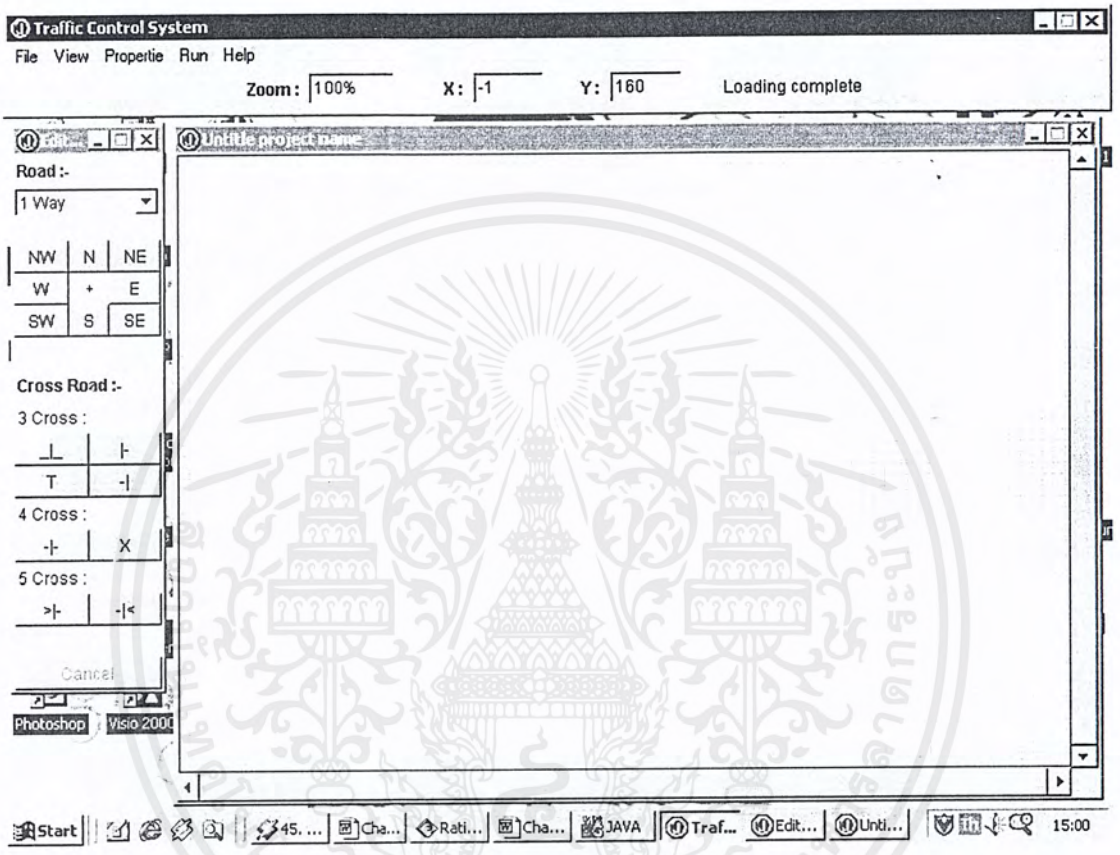
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลาส Controller เรียกใช้งานคลาส RunThread ซึ่งเป็นคลาสต้นแบบของการจำลองระบบสัญญาณไฟจราจร
4. คลาส RunThread ทำการสร้างวัตถุของตนเองขึ้นมา และกำหนดรายละเอียดของการวัตถุเอาไว้ด้วย
5. คลาส RunThread ส่ง Message บอกคลาส monitoring (เป็นคลาสที่ใช้ในการตรวจสอบการทำงานของสัญญาณไฟจราจร และรวมทั้งในส่วนของการแสดงผลการทำงานของระบบด้วย รายละเอียดจะกล่าวถึงในหัวข้อที่ 4.332)
6. คลาส Monitoring ทำการสร้างค่าต่างๆ เพื่อใช้ในการตรวจจับการทำงานของระบบสัญญาณไฟจราจร
7. คลาส RunThread ทำการคำนวณสัญญาณไฟจราจร (ข้อแตกต่างระหว่างการทำงานในแบบ Normal mode กับในแบบ AI mode นั้นต่างกันที่ส่วนนี้เท่านั้น)
8. คลาส RunThread ทำการส่งสัญญาณไฟจราจรทำคำนวณได้ไปให้คลาสของทางแยกในการปล่อยสัญญาณไฟจราจร
9. คลาสทางแยกส่งสัญญาณบอกส่วนของ monitoring ว่ามีการสั่งจังหวะของสัญญาณไฟจราจรเป็นอย่างไร และนานเท่าใด
10. คลาสทางแยกสั่งการเคลื่อนย้ายรถที่อยู่ในถนน
11. คลาสถนนทำการเคลื่อนย้ายรถที่อยู่ในตัวถนนเอง เคลื่อนย้ายรถไปยังถนนที่เชื่อมต่อกัน เคลื่อนย้ายรถไปยังปลายเปิด และรับรถจากถนนปลายเปิดที่ได้รับเข้ามาในถนน (รายละเอียดดูได้ที่หัวข้อ 3.2.3.6 คลาสถนน)
12. คลาสถนนทำการตรวจสอบจำนวนรถที่อยู่ในถนน เพื่อแสดงเป็นสี
13. คลาสของถนนแสดงผลสีของถนนที่ได้ออกทางหน้าจอ (เป็นส่วนของการทำงานใน thread ของการแสดงผล โดยจะทำงานควบคู่ไปกับ thread ของการจำลองระบบสัญญาณไฟจราจร)
14. เมื่อผู้ใช้สั่งการหยุดการจำลองระบบสัญญาณไฟจราจรผ่านทาง Interface หรืออาจจะครบตามกำหนดเวลาที่ผู้ใช้กำหนดเอาไว้
15. คลาส Controller ส่ง Message บอกคลาส RunThread ให้หยุดการจำลองระบบสัญญาณไฟจราจร
16. คลาส RunThread สั่งการหยุดสัญญาณไฟจราจร โดยจะต้องจบการทำงานของ thread ให้เรียบร้อย
17. คลาส RunThread ส่ง Message บอกคลาส Monitoring ให้ทำการแสดงผลรายละเอียดที่ได้เอาไว้
18. คลาส Monitoring ทำการคำนวณค่าต่างๆ ที่จำเป็นเพื่อใช้ในการแสดงผล และเพื่อใช้ในการประเมินการทำงานของระบบสัญญาณไฟจราจรที่ได้ออกมา
19. คลาส RunThread แสดงผลว่าการทำงานเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การออกแบบหน้าจอ (Screen Design)

การออกแบบหน้าจอของโปรแกรมถือว่าเป็นส่วนสำคัญมาก เพราะว่าหน้าจอของโปรแกรมจะเป็นส่วนผู้ใช้สามารถมองเห็นได้ และเป็นส่วนหลักที่ใช้ในการติดต่อกันระหว่างโปรแกรมกับผู้ใช้ การออกแบบหน้าจอในโครงงานนี้ มีการออกแบบหน้าจอให้ User Friendly มากที่สุด โดยจะมีลักษณะคร่าวของหน้าจอเป็นดังรูป



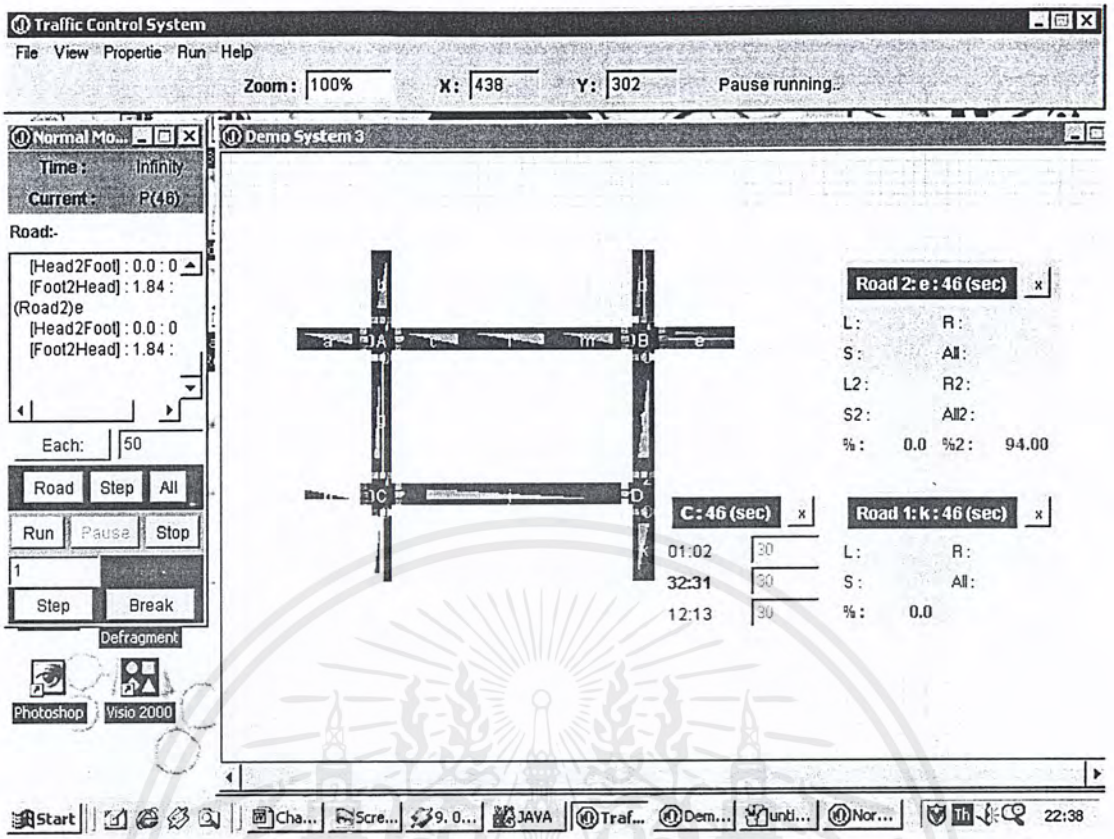
รูปที่ 3-34 แสดงหน้าจอคร่าวๆ ของโปรแกรม

จากรูปจะเห็นรูปร่างหน้าจอคร่าวๆ ของโปรแกรมโดยจะมี ส่วนประกอบที่สำคัญๆ อยู่ 3 ส่วนด้วยกัน คือ ส่วนของแผนที่, ส่วนของปุ่มในการควบคุมการทำงานของโปรแกรม (ส่วนด้านบนที่มีเมนูบาร์ในการควบคุมการทำงานเกี่ยวกับไฟล์ หรือจะเป็นการจำลองการทำงานของสัญญาณไฟจราจร) และในส่วนของการสร้างวัตถุลงในแผนที่เพื่อความสะดวกในการทำงานของโปรแกรม

ซึ่งจะมี Dialog เป็นส่วนที่ใช้ในการติดต่อกับผู้ใช้แบบชั่วคราวเพื่อใส่ค่า หรือทำงานบางอย่างที่ไม่ค่อยมีความสำคัญ

ส่วนของหน้าจอการแสดงผลในการจำลองการทำงานของสัญญาณไฟจราจรนั้นจะลักษณะของหน้าจอดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-35 แสดงลักษณะของหน้าจอการจำลองการทำงานของโปรแกรม

จากรูปด้านบน เราจะใช้สีในการแสดงจำนวนรถภายในถนนนั้นๆ ซึ่งแต่ละสีเราสามารถกำหนดเอาเองได้ว่าจะใช้สีไหนแทนปริมาณรถที่สนใจในโปรแกรม อีกทั้งยังสามรถใช้ Mouse เลือกในถนนที่สนใจแล้วแสดงปริมาณรถในถนนได้อีกด้วย และจะมีแถบควบคุมการทำงานด้านซ้ายเป็นส่วนของการควบคุมการทำงานของระบบจำลองสัญญาณไฟจราจร (จากรูปจะเห็นสัญญาณไฟจราจรของทางแยกด้วยว่ามีสัญญาณไฟจราจรที่ทางแยกเป็นอย่างไร)

3.2.6 การออกแบบ AI ที่ใช้ในการคำนวณสัญญาณไฟจราจร (AI Algorithm)

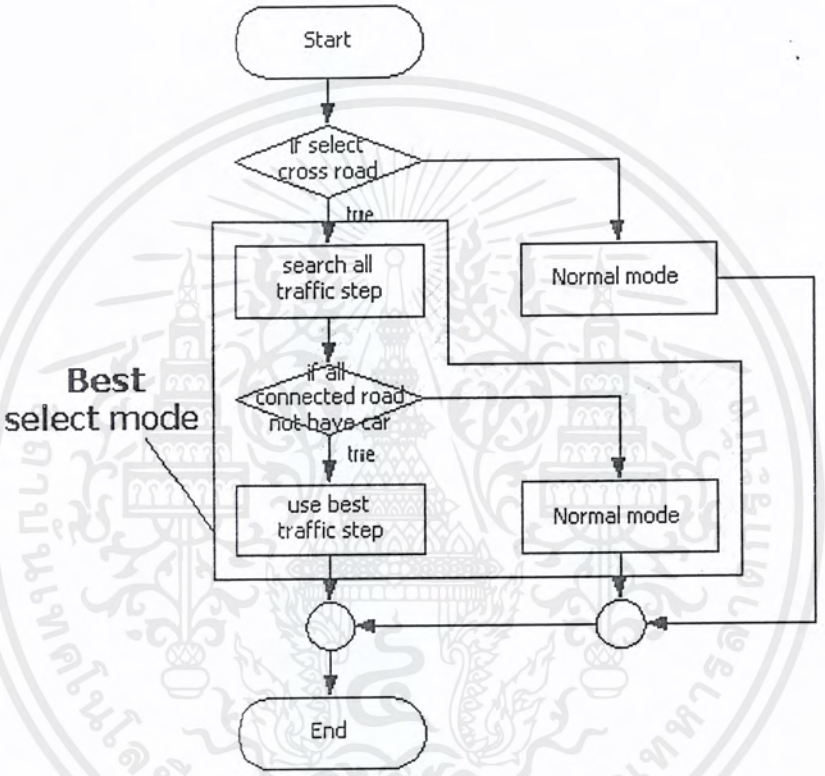
สำหรับเทคนิคทาง AI ที่นำมาใช้ในระบบสัญญาณไฟจราจรนั้นอาจจะไม่เหมือนกับเทคนิคทาง AI ของการทำงานอย่างอื่นมากนัก เพราะว่าต้องมีการประยุกต์การใช้งานมาใช้งานในการใช้งานจริงของระบบ ดังนั้นการใช้งานของระบบ AI ในสัญญาณไฟจราจรอาจจะสรุปได้ว่าเป็นการเลือกจังหวะของสัญญาณไฟจราจร และคำนวณเวลาในการทำงานให้เหมาะสม (เพราะว่าเราไม่สามารถควบคุมการวิ่งของรถในระบบได้ แต่เราควบคุมสัญญาณไฟจราจรในทางแยกได้) ซึ่งจากเทคนิคการทำงานทั้งสองอย่างที่ได้กล่าวมาแล้วนั้น เราสามารถขั้นตอนการคิดสัญญาณไฟจราจร ได้เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6.1 การคำนวณสัญญาณไฟจราจรแบบปล่อยปริมาณรถได้มากที่สุด (Best Select Mode)

การคำนวณแบบเลือกเส้นทางที่ดีที่สุดก่อนนั้นเปรียบเสมือนกับการที่มีตำรวจคอยควบคุมสัญญาณไฟจราจรที่ทางแยกนั่นเอง แต่การใช้ระบบคอมพิวเตอร์มาช่วยนั้นจะเป็นการทำงานที่ถูกต้องและแม่นยำมากยิ่งขึ้น (เนื่องจากตำรวจไม่สามารถรู้ว่ามีจำนวนรถที่ติดอยู่จริงเป็นเท่าใดในแต่ละทางแยก แต่ระบบในการทำงานของการคำนวณแบบเลือกเส้นทางที่ดีที่สุดนั้นมีการคิดเอาไว้ จึงทำให้การทำงานมีประสิทธิภาพมากกว่า แต่เป็นในทางทฤษฎีเท่านั้น ในทางปฏิบัติจริงนั้นอาจจะไม่ดีกว่าก็ได้)

การคำนวณสัญญาณไฟจราจรแบบเลือกเส้นทางที่ดีที่สุดก่อนนั้นมีการทำงานแบบคร่าวๆ ดังรูป



รูปที่ 3-36 แสดงการทำงานอย่างคร่าวๆ ของการคำนวณสัญญาณไฟแบบเลือกสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด

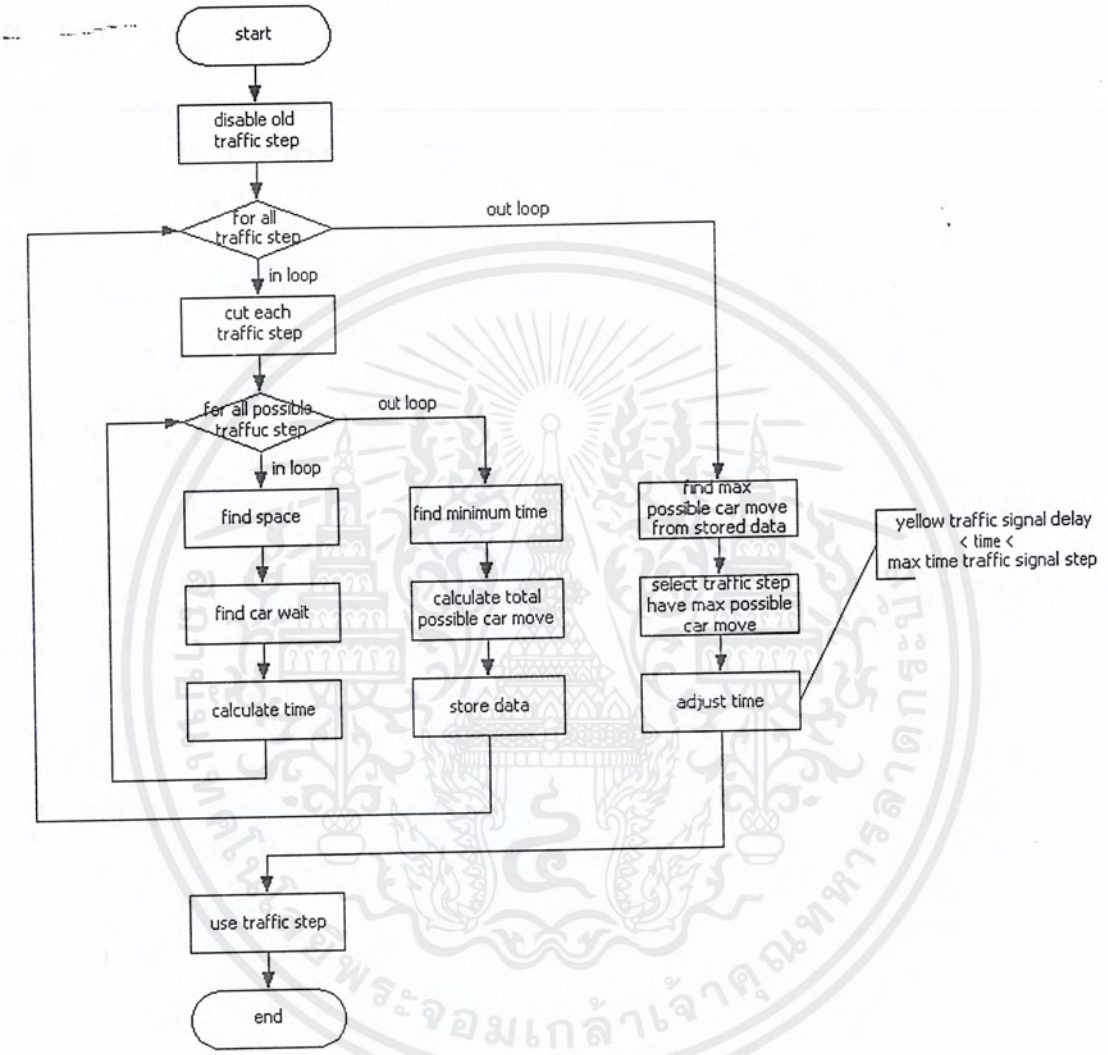
จากรูปจะเห็นส่วนของการทำงานเป็น 3 ส่วนคือ

1. เป็นส่วนของการทำงานของสัญญาณไฟจราจรแบบปรกติ คือเป็นการตั้งสัญญาณไฟจราจรแบบตามเวลาของการทำงาน (หรือในระบบจะเรียกว่า Normal Mode)
2. เป็นส่วนของการทำงานในส่วนของการเลือกจังหวะของสัญญาณไฟจราจรแบบที่ปล่อยรถมากที่สุด (หรือในระบบจะเรียกว่า Best Select Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการทำงานของสัญญาณไฟจราจรแบบปรกตินั้น ก็ไม่มีอะไรมาก เหมือนกับการทำงานของสัญญาณไฟจราจรโดยทั่วไป คือมีจังหวะ และเวลาในการปล่อยสัญญาณไฟจราจรที่แน่นอน

และในส่วนของการทำงานของสัญญาณไฟจราจรแบบเลือกเส้นทางที่ปล่อยปริมาณรถได้มากที่สุดนั้น แสดงการทำงานได้ดังรูป



รูปที่ 3-37 แสดงลักษณะของการคำนวณสัญญาณไฟจราจร
แบบเลือกสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด

ส่วนของการคำนวณสัญญาณไฟจราจรแบบเลือกสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด จะมีการทำงานดังนี้

1. หาจังหวะของสัญญาณไฟจราจรที่เป็นไปได้ โดยไม่นำสัญญาณไฟจราจรที่เคยใช้ไปแล้วมาคิด (เพราะว่าในความเป็นจริงไม่สามารถปล่อยจังหวะของสัญญาณไฟจราจรซ้ำกันได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หาเวลาที่เหมาะสมในการปล่อยสัญญาณไฟจราจร โดยพิจารณาจากพื้นที่ที่ถนนปลายทางรับได้ ปริมาณรถที่จะวิ่ง ในแต่ละหน่วยของสัญญาณไฟจราจร (เป็นรายละเอียดของจังหวะของสัญญาณไฟจราจร เช่น เลี้ยวซ้ายที่ถนน A เลี้ยวขวาที่ถนน B และวิ่งตรงไปของถนน C เป็นต้น) เมื่อได้เวลาของแต่ละหน่วยของสัญญาณไฟจราจรมาแล้วก็หาเวลาที่น้อยที่สุด เพื่อใช้ในการปล่อยสัญญาณไฟจราจรนั้นได้อย่างคุ้มค่านั่นเอง ความสัมพันธ์ของปริมาณรถกับเวลาเป็นดังนี้

Time = Car / Car Rate

เมื่อ

Time เป็นเวลาของสัญญาณไฟจราจรที่ปล่อยได้ มีหน่วยเป็นวินาที

Car เป็นปริมาณรถที่วิ่งได้ ซึ่งจะเป็นปริมาณพื้นที่ว่างของถนนที่จะรับ หรืออาจจะเป็นปริมาณรถที่จะวิ่งก็ได้ โดยจะใช้ค่าที่น้อยที่สุดมาคิด

Car Rate เป็นปริมาณรถที่วิ่งได้ในถนนนั้นจริงๆ เป็นค่าคงที่ที่หามาได้จากหัวข้อที่ 2.4.4 การปล่อยรถที่ทางแยก

3. เลือกจังหวะของสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุดก่อน
4. ปรับค่าของเวลาที่ได้โดยค่าของเวลาที่จะสั่งสัญญาณไฟจราจรนั้นจะต้องเป็นดังนี้

ระยะหน่วงเวลาของสัญญาณไฟเหลือง ≤ ระยะเวลาที่ปล่อยสัญญาณไฟ ≤ ระยะเวลามากสุดที่ปล่อยสัญญาณไฟนั้นได้ (เป็นค่าที่ตั้งโดยผู้ใช้)

เพราะว่าในการทำงานจริงๆ นั้นอาจจะมีเวลาที่ปล่อยมากๆ ได้ (ในกรณีที่รถติดมากๆ) และอาจจะมีกรณีที่ปล่อยสัญญาณไฟจราจรน้อยมากๆ ได้เช่นกัน (ในกรณีที่มีรถน้อยๆ แต่ไม่เกิดปัญหาถนนว่าง)

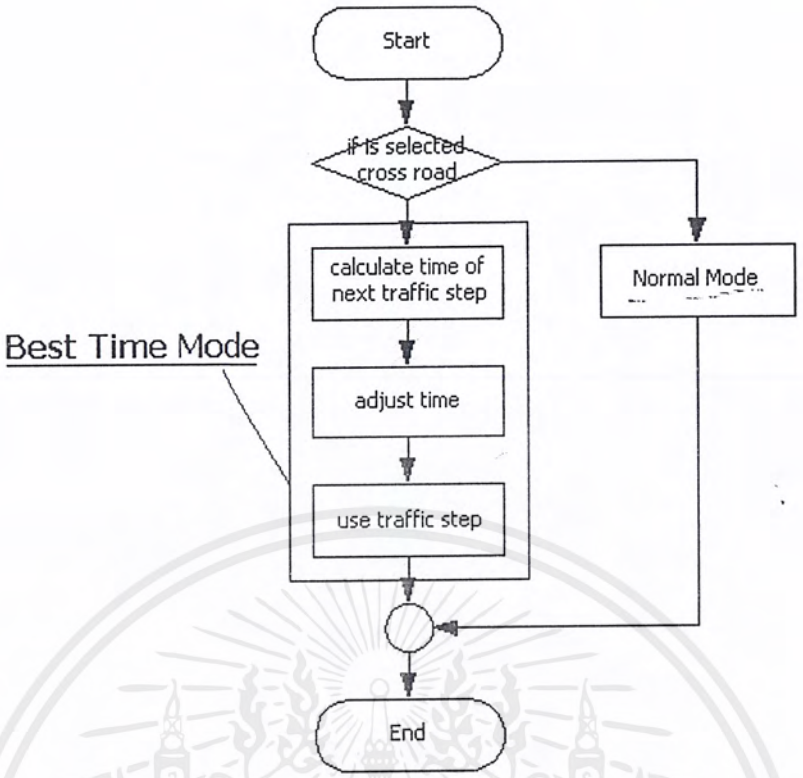
3.2.6.2 การคำนวณสัญญาณไฟแบบปล่อยสัญญาณไฟจราจรให้คุ้มค่าที่สุด

การคำนวณสัญญาณไฟจราจรแบบนี้นั้นมีลักษณะของการทำงานที่คล้ายคลึงกับการทำงานของสัญญาณไฟจราจรแบบปรกติ (สัญญาณไฟจราจรแบบตั้งเวลาการทำงานเอาไว้) และส่วนของการทำงานแบบเลือกสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด โดยมีรายละเอียดเป็น

ส่วนของการเลือกจังหวะของสัญญาณไฟจราจรนั้นจะเลือกสัญญาณไฟจราจรแบบปรกติ คือมีลำดับการทำงานของสัญญาณไฟจราจรที่แน่นอน

ส่วนของการหาเวลาในการปล่อยสัญญาณไฟจราจร จะมีการคำนวณเวลาการทำงานเช่นเดียวกับการทำงานของสัญญาณไฟจราจรแบบเลือกสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุด คือมีการหาพื้นที่ว่างของถนนที่จะรับ และหาปริมาณรถที่จะวิ่งว่ามีปริมาณเท่าใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-38 แสดงลักษณะของการคำนวณสัญญาณไฟจราจร
แบบหาเวลาที่ดีที่สุดในการปล่อยสัญญาณไฟจราจร

จากรูปจะเห็นว่ามีการทำงานคำนวณเวลาของจังหวะสัญญาณไฟจราจรเฉพาะจังหวะของสัญญาณไฟจราจรถัดไปเท่านั้น ไม่มีการหาทั้งหมดเหมือนการคำนวณสัญญาณไฟจราจรแบบเลือกจังหวะของสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุด

ในส่วนของการปรับเวลาที่ใช้ในการปล่อยสัญญาณไฟจราจรให้มีค่าเป็นดังนี้

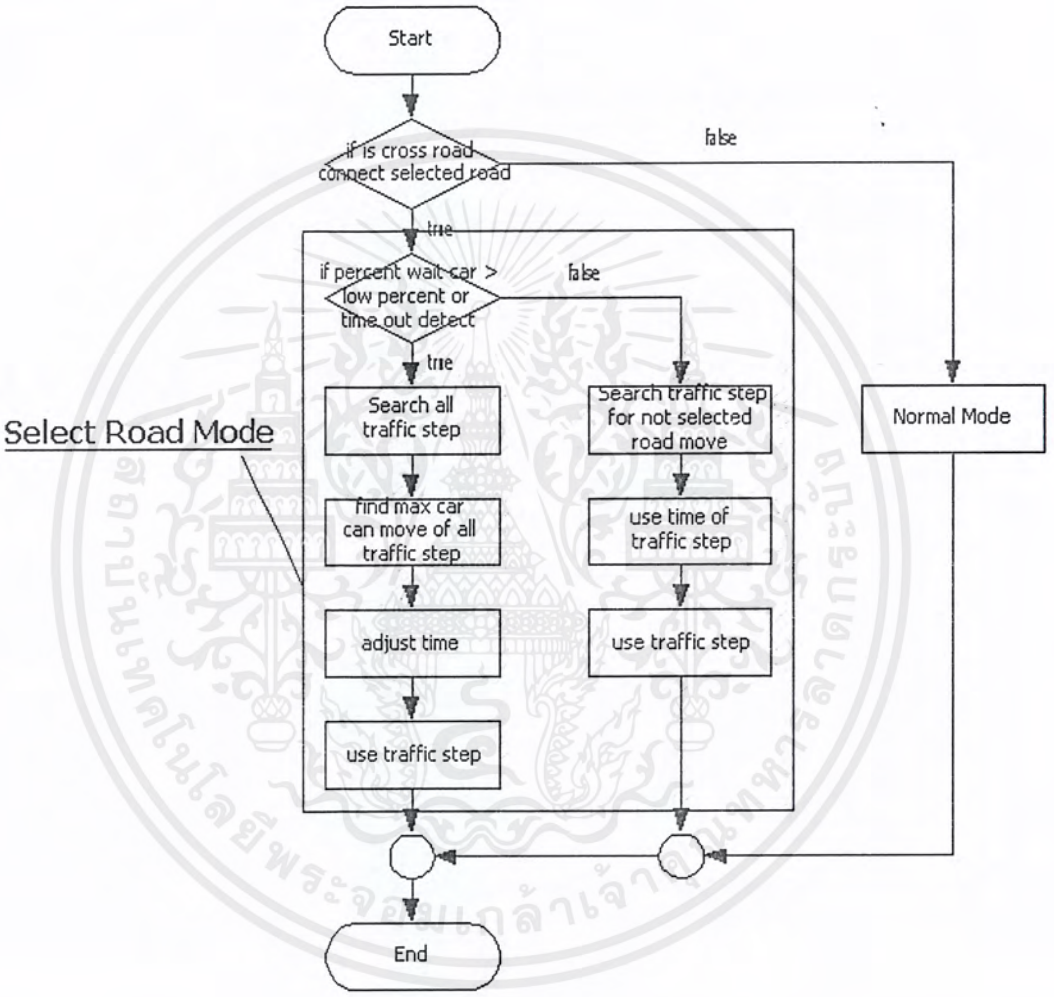
ระยะหน่วงเวลาของสัญญาณไฟเหลือง $\leq$ เวลาที่ปล่อยสัญญาณไฟจราจร $\leq$ ระยะเวลามากที่สุดที่สามารถปล่อยจังหวะสัญญาณไฟจราจรนั้นได้

จะเห็นได้ว่ามีการปรับเวลาของการปล่อยสัญญาณไฟจราจรเป็นเช่นเดียวกับการคำนวณสัญญาณไฟจราจรแบบเลือกสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุดด้วยเหตุผลเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6.3 การคำนวณสัญญาณไฟจราจรแบบเลือกถนนที่ต้องการให้รถติดน้อยที่สุด

การคำนวณสัญญาณไฟจราจรแบบนี้นั้นผู้ใช้สามารถกำหนดได้ว่าต้องการจะให้ถนนเส้นใดมีปริมาณรถในถนนเท่าใดโดยทางแยกที่เชื่อมต่อกับถนนนั้นๆ จะปล่อยสัญญาณไฟจราจรเพื่อช่วยให้รถในถนนเส้นนั้นมีปริมาณรถในถนนใกล้เคียงกับที่เราต้องการที่สุด โดยผู้ใช้สามารถเลือกถนนที่ต้องการขึ้นมาและกำหนดว่าต้องการให้ถนนที่เลือกขึ้นมานั้นมีปริมาณรถติดต่อปริมาณรถที่ถนนรับได้เท่าไร ซึ่งผู้ใช้นั้นจะสามารถป้อนค่าได้เป็นเปอร์เซ็นต์ของปริมาณรถในถนนเส้นนั้นได้



รูปที่ 3-39 แสดงลักษณะของการคำนวณสัญญาณไฟจราจรแบบเลือกถนนที่ต้องการให้รถติดน้อยที่สุด

การคำนวณสัญญาณไฟจราจรในแบบนี้นั้นจะมีอยู่ 2 ส่วนคือ

- ส่วนแรก เป็นส่วนของการป้อนค่าลงไป โดยที่ผู้ใช้สามารถเลือกถนนที่ต้องการ และเลือกเปอร์เซ็นต์ของปริมาณรถที่ติดรอสัญญาณไฟอยู่ในถนนนั้น (จะมีค่าตั้งต้นที่ใช้เป็น 30%)
- ส่วนที่สอง เป็นส่วนของการคำนวณสัญญาณไฟจราจรในโหมดการทำงานนี้ จากรูปที่แสดงนั้น จะมีขั้นตอนการทำงานเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าค่าเปอร์เซ็นต์ของปริมาณรถที่ติดรอสัญญาณไฟอยู่ในถนนที่สนใจนั้น มีค่ามากกว่าที่ผู้ใช้กำหนดไว้ หรือถนนนั้นรอนานเกินเวลาที่กำหนดเอาไว้ ก็จะทำการปล่อยสัญญาณไฟจราจรเพื่อลดปริมาณรถในถนนที่สนใจนั้นลง เพื่อทำให้เปอร์เซ็นต์ของรถที่ติดในถนนนั้นลดลง โดยใช้การหาจังหวะของสัญญาณไฟที่ดีที่สุดในการระบายรถออกจากถนนนั้นได้มากที่สุด (จะมีการคำนวณสัญญาณไฟจราจรเช่นเดียวกับการคำนวณสัญญาณไฟจราจรแบบเลือกสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุด แต่มีการสนใจปริมาณที่วิ่งได้เป็นส่วนของรถที่อยู่ในถนนที่สนใจแทนที่จะเป็นปริมาณรถที่วิ่งอยู่ทั้งถนนแทน) แต่ถ้าค่าเปอร์เซ็นต์ของปริมาณรถที่ติดรอสัญญาณไฟอยู่ในถนนที่สนใจนั้น มีค่าน้อยกว่าที่เรากำหนดไว้ จะทำการปล่อยรถในถนนเส้นอื่นๆ แทน โดยอาศัยการปล่อยรถแบบเลือกสัญญาณไฟจราจรที่ไม่มีส่วนของสัญญาณไฟจราจรที่ปล่อยรถในถนนที่เลือก (เพื่อเป็นการระบายรถในเส้นทางอื่น ให้รถติดน้อยลง)

ในการทำงานของสัญญาณไฟจราจรในแบบนี้นั้นจะไม่สนใจถนนอื่นๆ เลย แต่จะสนใจปริมาณรถในถนนที่สนใจเท่านั้น ซึ่งในการใช้งานนั้นก็จะพบเห็นการใช้งานได้น้อยมาก แต่ว่าก็จะมีการใช้ในกรณีเช่นในถนนนั้นจำเป็นต้องมีการใช้งาน เช่นเป็นถนนที่ในหลวงจะเสด็จผ่าน เป็นต้น จึงจำเป็นจะต้องระบายรถออกจากถนนนั้นออกไป แต่การทำงานที่จะเกิดประสิทธิภาพนั้นจำเป็นจะต้องอาศัยการควบคุมทางแยกหลายๆ ทางแยกด้วยกันเพื่อกระจายปริมาณรถออกไปจากระบบให้มากที่สุด จึ่งจะเป็นการทำงานที่มีประสิทธิภาพนั่นเอง

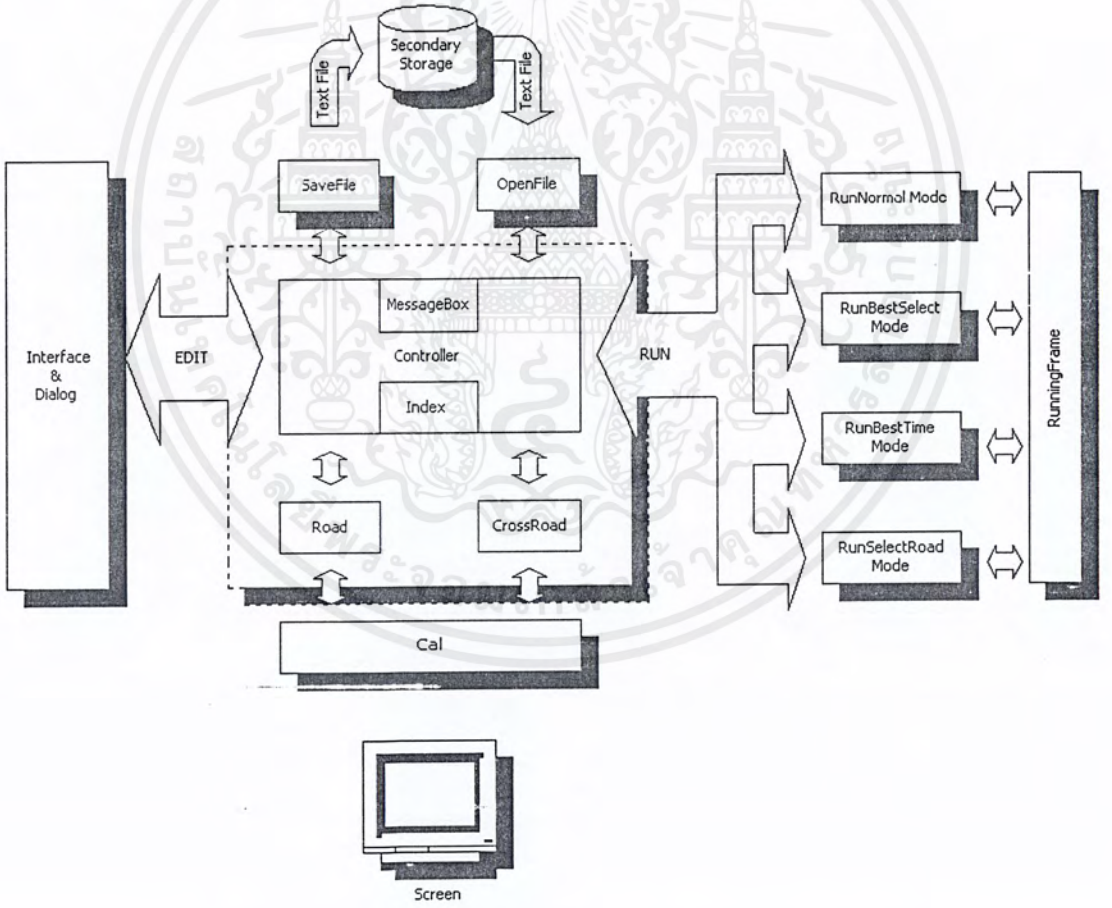
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การเขียนโปรแกรม (Implementation)

การเขียนโปรแกรมเป็นการทดสอบการทำงานของโปรแกรมที่เราได้ออกแบบเอาไว้ ซึ่งเป็นส่วนสำคัญในการทำงาน การออกแบบของโปรแกรมจะดีหรือไม่ดีนั้น เมื่อเราทำการเขียนโปรแกรมแล้วจึงจะทราบได้ว่าโปรแกรมที่ได้ออกแบบนั้นดีหรือไม่ดีอย่างไร

4.1 อธิบายคลาสต่างๆ ที่ใช้ในโปรแกรม

คลาสที่ใช้ในโปรแกรมจะเขียนโปรแกรมแบ่ง 1 ไฟล์ต่อ 1 คลาส และมีการแบ่งไฟล์ต่างๆ ออกเป็น Package เพื่อความง่ายต่อการนำไปใช้และการนำกลับมาใช้ใหม่ การแบ่ง Package ของการทำงานของโปรแกรมนั้น แสดงการแบ่ง Package ต่างๆ ในโปรแกรมได้ดังรูป



รูปที่ 4-1 แสดง Package ของคลาสต่างๆภายในโปรแกรมอย่างคร่าวๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเราสามารถแบ่ง Package ของการทำงานได้ออกเป็นดังนี้

ชื่อของ Package	การทำงานภายในโปรแกรม
TrafficControlSystem.kernal	เป็นส่วนแกนหลักของการทำงานของโปรแกรม ประกอบไปด้วย คลาส Controller, CrossRoad, Cross3, Cross4, Cross5, Road, Road1, Road2, TrafficLight, Index และ MessageBox
TrafficControlSystem.io	เป็นส่วนของการ บันทึกและการเปิดไฟล์ ซึ่งประกอบไปด้วยคลาส SaveFile และ OpenFile
TrafficControlSystem.display	เป็นส่วนของการแสดงผลของโปรแกรมออกทางหน้าจอ ซึ่งมีคลาสเดียวคือ คลาส Cal
TrafficControlSystem.ui	เป็นส่วน Interface ของโปรแกรมทั้งหมด และเป็นส่วนที่สำคัญมากในการแลดงผลของโปรแกรมออกทางหน้าจอ ประกอบไปด้วยคลาส AboutDialog, BaseDialog, crossConnectDialog, EditorFrame, HelpDialog, Interface, MapFrame, ShowTrafficStep, MouseMotionListen, ObjectDialog, PropertiesCrossDialog, PropertiesProgramDialog, PropertiesProjectDialog, ShowRoad, PropertiesRoadDialog, RoadDialog, RunAIDialog, RunningFrame, RunNormalDialog และ SplashScreen
TrafficControlSystem.ui.io	เป็นส่วนของการบันทึกค่าของการ Run โปรแกรมเอาไว้ในไฟล์ ซึ่งมีคลาสเพียงคลาสเดียวคือ SaveOutput
TrafficControlSystem.run	เป็นส่วนของการ Run โปรแกรม เป็นส่วนของการทำงานที่แยกออกจากส่วนของ kernel มีคลาสคือ RunThread, RunNormalThread, RunBestSelectThread, RunBestTimeThread และ RunSelectRoadThread
TrafficControlSystem.run.ui	เป็นส่วนของ Interface ที่ใช้ติดต่อกับผู้ใช้ของ Package TrafficControlSystem.run มีคลาสคือ CrossDialog, LowPercentDialog และ RoadDialog

ตารางที่ 4-1 แสดง Package ต่างๆ ของโปรแกรม

จากตารางด้านบน จะเห็นว่ามี Package ต่างๆ อยู่หลาย Package ดังนั้นเราจะมาดูรายละเอียดของแต่ละ Package ตามลำดับดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 Package kernel

เป็น Package หลักของการทำงานของโปรแกรม อาจจะกล่าวได้ว่าเป็นส่วนที่สำคัญที่สุด เพราะว่า Package อื่นในโปรแกรมจะต้องใช้และติดต่อกับ Package นี้เสมอ มีรายละเอียดของคลาสต่างๆ ดังนี้

4.1.1.1 คลาส Controller

เป็นคลาสที่สำคัญที่สุดของการโปรแกรม เพราะเป็นส่วนที่ควบคุมการทำงานทั้งหมดเอาไว้ (รายละเอียดของคลาส ดูได้จากหัวข้อที่ 3.2.3.1 คลาส Controller)

4.1.1.2 คลาส Road

เป็นคลาสต้นแบบของถนนทั้งหมดในระบบ อีกทั้งยังเป็นส่วนที่เก็บค่าของรถที่อยู่ในระบบ ซึ่งเป็นค่าที่เราสนใจมาก และต้องการให้มันมีค่าน้อยที่สุดเท่าที่เป็นไปได้ด้วย (รายละเอียดของคลาสดูได้จากหัวข้อที่ 3.2.3.6 คลาสถนน)

4.1.1.3 คลาส Road1

เป็นคลาสของถนนชนิดทางเดียว เป็นคลาสที่ขยายมาจากคลาส Road และมีการ Implement method ของคลาสต้นแบบเอาไว้ ซึ่งเราจะใช้วัตถุของคลาสนี้ในระบบ (รายละเอียดดูได้จากหัวข้อที่ 3.2.3.7 คลาสถนนทางเดียว)

4.1.1.4 คลาส Road2

เป็นคลาสของถนนสองทาง เป็นคลาสที่ขยายมาจากคลาส Road เหมือนกันกับคลาส Road1 แต่ใช้สร้างวัตถุของถนนสองทาง และมีการ Implement method ของคลาส Road ด้วย (รายละเอียดดูได้จากหัวข้อที่ 3.2.3.8 คลาสถนนสองทาง)

4.1.1.5 คลาส CrossRoad

เป็นคลาสต้นแบบของทางแยกทั้งหมดในระบบ โดยทางแยกทุกทางแยกจะมีการทำงานที่คล้ายคลึงกัน ดังนั้นจึงสร้างคลาสต้นแบบมาเพื่อจัดการ และสามารถเรียกใช้งานได้เหมือนกันทั้งระบบ (รายละเอียดสามารถดูได้จากหัวที่ 3.2.3.2 คลาสทางแยก)

4.1.1.6 คลาส Cross3

เป็นคลาสของสามแยก มีการสืบทอดมาจากคลาส CrossRoad และมีการ Implement method ในคลาส CrossRoad ด้วย โดยระบบจะใช้วัตถุของคลาสนี้เป็นสามแยกในระบบ (รายละเอียดดูได้จากหัวข้อที่ 3.2.3.3 คลาสสามแยก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.7 คลาส Cross4

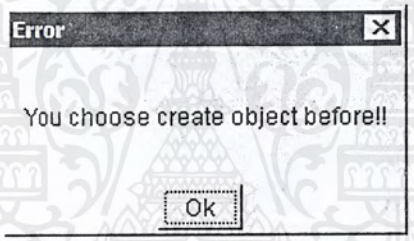
เป็นคลาสของสี่แยกในระบบ มีการสืบทอดมาจากคลาส CrossRoad เช่นเดียวกับคลาส Cross3 ซึ่งในระบบจะใช้วัตถุของคลาสนี้เป็นสี่แยกในระบบการทำงาน (รายละเอียดดูได้จากหัวข้อที่ 3.2.3.4 คลาสสี่แยก)

4.1.1.8 คลาส Cross5

เป็นคลาสของห้าแยกในระบบ มีการสืบทอดมาจากคลาส CrossRoad เช่นเดียวกับคลาส Cross3 และคลาส Cross4 โดยระบบจะใช้วัตถุของคลาสนี้เป็นวัตถุห้าแยกในระบบ (รายละเอียดของคลาสสามารถดูได้ที่หัวข้อ 3.2.3.5 คลาสห้าแยก)

4.1.1.9 คลาส MessageBox

เป็นคลาสของ Dialog ที่ใช้ติดต่อกับผู้ใช้ เพื่อแสดงข้อความต่างๆ ของระบบเป็นส่วนหน้าจอการแสดงผลของระบบ มีลักษณะดังรูป



รูปที่ 4-1 แสดงลักษณะของคลาส MessageBox ในการข้อความ

4.1.1.10 คลาส Index

เป็นคลาสที่ใช้ในการหาค่าของ index ของวัตถุใน Controller และยังเป็นส่วนของอรรถประโยชน์ต่างๆ ของระบบ (รายละเอียดดูได้ที่หัวข้อ 3.2.3.12 คลาส Index)

4.1.1.11 คลาส TrafficLight

เป็นคลาสของสัญญาณไฟจราจรในระบบ เป็นส่วนของสัญญาณไฟจราจรเพียงอย่างเดียว โดยมีลักษณะของสัญญาณไฟแบบ Full Access คือมีทั้งเลี้ยวซ้าย เลี้ยวขวา และตรงไปอยู่ในอันเดียวกัน (รายละเอียดของคลาสดูได้ที่หัวข้อ 3.2.3.11 คลาส TrafficLight)

4.1.2 Package display

เป็น Package ที่ใช้ในการแสดงผลของวัตถุออกทางหน้าจอ และยังใช้ในการคำนวณการแสดงผลของวัตถุในระบบออกทางแผนที่อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.1 คลาส Cal

เป็นคลาสที่ใช้ในการแสดงผลของระบบการจราจร โดยจะเป็นการช่วยในการคำนวณจุดต่างๆ ที่อยู่ในวัตถุนั้นๆ (รายละเอียดดูได้ที่หัวข้อที่ 3.2.3.10 คลาส Cal)

4.1.3 Package io

เป็น Package การทำงานที่เกี่ยวกับส่วนของการบันทึกแผนที่ และการเปิดแผนที่ของระบบสัญญาณไฟจราจร โดยจะมีคลาสภายใน Package นี้อยู่ 2 คลาสดังนี้

4.1.3.1 คลาส SaveFile

เป็นคลาสในส่วนของการบันทึกแผนที่ของระบบ โดยจะถูกเรียกใช้งานผ่านทางคลาส Controller อีกทอดหนึ่ง โดยจะบันทึกไฟล์เป็นไฟล์นามสกุล ".trf" (รายละเอียดของคลาสนั้นดูได้ที่หัวข้อที่ 3.2.3.9 คลาส SaveFile)

4.1.3.2 คลาส OpenFile

เป็นคลาสการทำงานในส่วนของการเปิดแผนที่ของระบบ โดยจะถูกเรียกใช้งานผ่านทางคลาส Controller เช่นเดียวกันกับคลาส SaveFile โดยจะเปิดไฟล์นามสกุล ".trf" เท่านั้น (รายละเอียดของคลาสดูได้ที่หัวข้อ 3.2.3.8 คลาส OpenFile)

4.1.4 Package Run

เป็น Package การทำงานที่สำคัญ เพราะเป็นการคำนวณสัญญาณไฟจราจรและนำมาใช้ในระบบที่ได้ออกแบบเอาไว้ โดยจะมีคลาสในการทำงานเป็นดังนี้

4.1.4.1 คลาส RunThread

เป็นคลาสต้นแบบของการทำงานของทุกคลาสที่ต้องการจำลองระบบสัญญาณไฟจราจรในระบบการทำงานของคลาสนี้ ซึ่งรายละเอียดของคลาสนี้สามารถดูได้จากหัวข้อที่ 3.2.3.13 คลาส RunThread)

4.1.4.2 คลาส RunNormalThread

เป็นคลาสที่จำลองระบบสัญญาณไฟจราจรแบบปรกติ คือมีการตั้งจังหวะของสัญญาณไฟจราจรตามเวลาที่ได้กำหนดเอาไว้ ซึ่งรายละเอียดดูได้ที่หัวข้อ 3.2.3.14 คลาส RunNormalThread

4.1.4.3 คลาส RunBestSelectThread

เป็นคลาสที่มีการคำนวณสัญญาณไฟจราจรแบบปล่อยรถให้วิ่งออกไปมากที่สุด การทำงานของคลาสนี้จะเป็นการทำงานในโหมดของ AI เช่นเดียวกับของคลาส RunBestTimeThread และ RunSelectRoadThread โดยที่รายละเอียดของการออกแบบการคำนวณนั้นดูได้จากหัวข้อที่ 3.2.6.1 การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบสัญญาณไฟจราจรให้ปล่อยรถออกมาได้มากที่สุด และในส่วนของการทำงานของคลาสนั้นดูได้ที่หัวข้อ 3.2.3.16 คลาส RunBestSelectThread

4.1.4.4 คลาส RunBestTimeThread

เป็นคลาสที่จำลองสัญญาณไฟจราจรโดยใช้การคำนวณเวลาของสัญญาณไฟเพื่อให้ปล่อยสัญญาณไฟได้ดีที่สุด โดยจะไม่มีการเลือกว่าจะปล่อยสัญญาณไฟใด (เป็นการลำดับสัญญาณไฟจราจรไปเรื่อยๆ แต่มีการคำนวณเวลา) ซึ่งรายละเอียดของคลาสสามารถดูได้ที่หัวข้อ 3.2.3.17 คลาส RunBestTimeThread ส่วนรายละเอียดของการออกแบบการคำนวณสัญญาณไฟจราจรดูได้จากหัวข้อที่ 3.2.6.2 การคำนวณสัญญาณไฟจราจรแบบเลือกเวลาที่ดีที่สุด

4.1.4.5 คลาส RunSelectRoadThread

เป็นคลาสการจำลองการทำงานของสัญญาณไฟจราจรโดยใช้การคำนวณเวลาของสัญญาณไฟจราจรและเลือกสัญญาณไฟจราจรเพื่อให้ปล่อยรถออกจากถนนที่เลือกได้มากที่สุด (เป็นการเลือกสัญญาณไฟจราจรและคำนวณเวลาในการปล่อยสัญญาณไฟจราจร) ซึ่งรายละเอียดของการคำนวณสัญญาณไฟจราจรสามารถดูได้ที่หัวข้อ 3.2.6.3 การคำนวณสัญญาณไฟจราจรแบบเลือกเส้นทางที่จะปล่อยรถให้ได้มากที่สุด ส่วนรายละเอียดของคลาสสามารถดูได้ที่หัวข้อ 3.2.3.18 คลาส RunSelectRoadThread

4.1.4.6 Sub Package ui

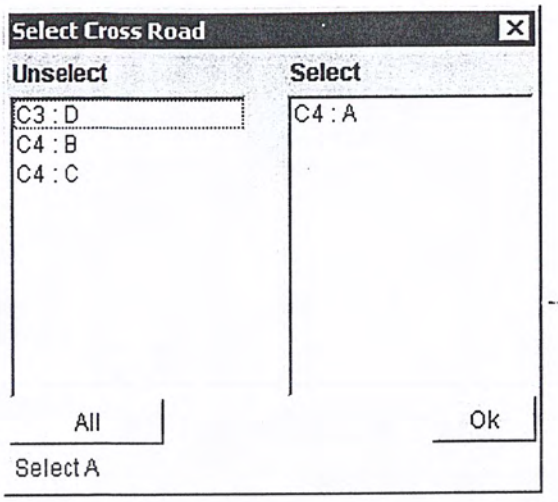
เป็น Package ที่เก็บหน้าจอของการติดต่อผู้ใช้ในส่วนของการเลือกถนนหรือเลือกทางแยกที่ต้องการคำนวณระบบสัญญาณไฟจราจร ซึ่งจะมีคลาสที่ใช้ในการทำงานอยู่ 3 คลาสดังนี้

4.1.4.6.1 คลาส CrossDialog

เป็น Dialog ที่ใช้ในการเลือกทางแยก (ใช้ในส่วนของการเลือกทางแยกของการคำนวณสัญญาณไฟจราจรแบบปล่อยจังหวะที่รถวิ่งออกไปได้มากที่สุดก่อน) มีลักษณะเป็นดังรูปที่ 4-2

จากรูปที่ 4-2 จะแบ่งการทำงานหน้าจอออกเป็น 2 ส่วน คือ ส่วนของทางแยกที่เลือก และส่วนของทางแยกที่ไม่ได้เลือก โดยจะอยู่คนละฝั่งของ List ที่แสดง ทางแยกที่เลือกจะอยู่ที่ List ทางขวามือ (Select) ส่วนของทางแยกที่ไม่ได้เลือกจะอยู่ทางซ้ายมือ (Unselect) สามารถเลือกได้โดยดับเบิลคลิกเมาส์ของ List เพื่อใช้ในการเลือกทางแยก (โดยจะมีแถบการเลือกบ่งบอกอยู่ด้านล่าง)

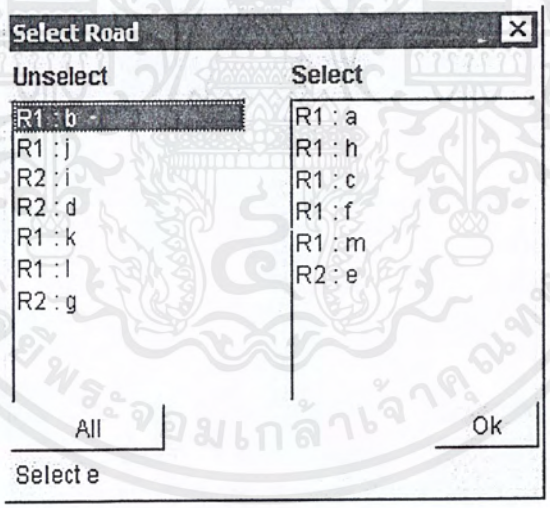
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-2 แสดงลักษณะของหน้าจอของคลาส CrossDialog

4.1.4.6.2 คลาส RoadDialog

เป็นคลาสที่ให้ผู้ใช้เลือกถนนที่ต้องการ (เรียกใช้งานจากคลาส RunSelectRoadThread) ซึ่งจะมีลักษณะของหน้าจอแสดงได้ดังรูป



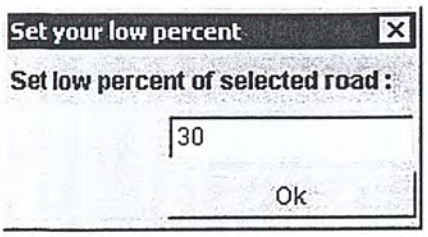
รูปที่ 4-3 แสดงลักษณะของหน้าจอของ RoadDialog

ลักษณะของการใช้งานเหมือนกันกับคลาส CrossDialog แต่ว่าเป็นการเลือกถนนแทนที่จะเป็นการเลือกทางแยก

4.1.4.6.3 คลาส LowPercentDialog

เป็น Dialog ที่ให้ผู้ใช้กำหนดเปอร์เซ็นต์ของรถในถนนที่สูงที่สุดที่เป็นไปได้ ใช้ในการทำงานในโหมดของการปล่อยสัญญาณไฟจราจรแบบเลือกถนนที่ติดน้อยที่สุด มีลักษณะของ Dialog เป็นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-4 แสดงลักษณะหน้าจอของ LowPercentDialog

4.1.5 Package monitoring

เป็น Package ที่ใช้ในการตรวจวัดผลของการทำงานของการจำลองสัญญาณไฟจราจรที่ได้ โดยจะมีคลาสภายใน Package นี้อยู่ 1 คลาสดังนี้

4.1.5.1 คลาส Monitoring

เป็นคลาสที่ใช้ในการแสดงผลของการจำลองระบบสัญญาณไฟจราจร โดยจะเก็บข้อมูลต่างๆ ของถนนและทางแยก รวมทั้งยังใช้ในการคำนวณสมรรถนะของการทำงานของระบบสัญญาณไฟจราจรด้วย ว่ามีสมรรถนะในการทำงานเป็นอย่างไร เพื่อใช้ในการวิเคราะห์และประเมินผลการทำงานที่ได้เพื่อนำไปพัฒนาต่อไป

การใช้งานคลาสนี้นั้นจะมีการใช้อยู่ภายในตัวโปรแกรมเลย และมีการฝังการตรวจจับเอาไว้ในวัตถุด้วย เพราะฉะนั้นในการจำลองการทำงานของระบบสัญญาณไฟจราจรนั้นจะต้องมีสืบทอดมาจากคลาส RunThread อีกทอดหนึ่ง (เพราะว่าภายในคลาส RunThread มีการกำหนดค่าเริ่มต้นของ Monitoring นี้เอาไว้แล้ว) โดยที่ในโปรแกรมจะมีการเรียกใช้คลาสนี้ที่ส่วนของการแสดงผลออกทางหน้าจอในขณะที่ทำการจำลองระบบสัญญาณไฟจราจรเลย และเรียกใช้ในตอนบันทึกข้อมูลลงไปในไฟล์ถ้าผู้ใช้ต้องการบันทึกค่า

ค่าที่สนใจในการตรวจสอบจะมีดังนี้

1. ปริมาณรถที่อยู่ในถนน ณ วินาทีนั้นๆ
2. เปอร์เซ็นต์ของรถที่อยู่ในถนนนั้น
3. ปริมาณรถที่เข้าสู่ถนนนั้นๆ
4. ปริมาณรถที่ออกจากถนนนั้นๆ
5. จังหวะของสัญญาณไฟจราจรที่เปลี่ยน
6. ค่าอัตราการวิ่งของถนน

จากค่าที่สนใจนี้ค่าของอัตราการวิ่งของถนนนั้นจะเป็นค่าที่เราสนใจมาก เพราะว่าเป็นค่าที่ใช้ในการบอกประสิทธิภาพของการทำงานของระบบสัญญาณไฟจราจรนั้นๆ ว่ามีประสิทธิภาพของการทำงานเป็นอย่างไร

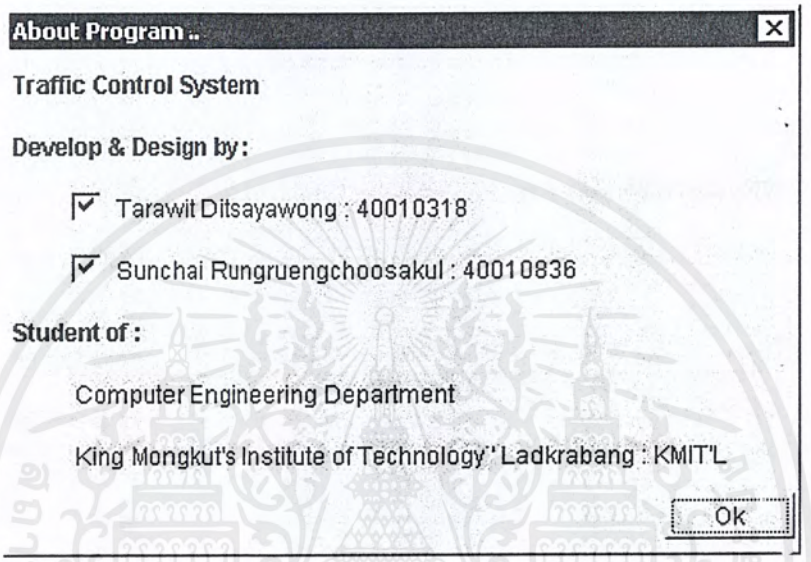
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 Package ui

เป็น Package ที่เกี่ยวข้องกับหน้าจอของโปรแกรมทั้งหมด โดยจะเป็นส่วนของการติดต่อกับผู้ใช้แบบ Graphic User Interface (GUI) ทั้งหมด โดยมีรายละเอียดของคลาสเป็นดังนี้

4.1.6.1 คลาส AboutDialog

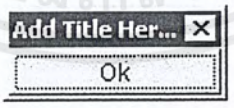
เป็นคลาสที่แสดงถึงรายชื่อของผู้ออกแบบและพัฒนาโปรแกรมนี้ มีลักษณะดังรูป



รูปที่ 4-5 แสดงหน้าจอของคลาส AboutDialog

4.1.6.2 คลาส BaseDialog

เป็นคลาสต้นแบบของ Dialog ของโปรแกรมทั้งหมด จะเป็น Dialog แบบ Modal (เป็น Dialog ที่ผู้ใช้ต้องตอบรับก่อนจึงจะทำงานต่อไปได้) มีลักษณะของหน้าจอเป็นดังรูป

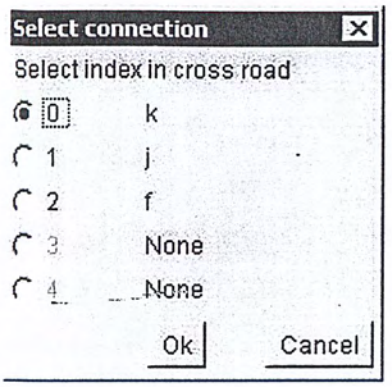


รูปที่ 4-6 แสดงหน้าจอของ BaseDialog

4.1.6.3 คลาส crossConnectDialog

เป็น Dialog ที่ติดต่อกับผู้ใช้ โดยให้ผู้ใช้เลือก index ของถนนในทางแยกว่าต้องการติดต่อที่ส่วนใด โดยจะถูกเรียกใช้งานโดยคลาส PropertiesRoadDialog เมื่อเลือกการเชื่อมต่อของถนนที่ด้าน Head หรือด้าน Foot เข้ากับทางแยก ลักษณะของหน้าจอเป็นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

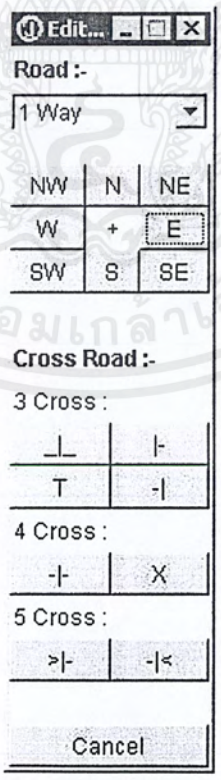


รูปที่ 4-7 แสดงหน้าจอของคลาส crossConnectDialog

Label ที่แสดงเป็นส่วนของถนนที่เชื่อมต่อกับทางแยกอยู่ในขณะนั้น และส่วนของ Radio Button ใช้สำหรับเลือก index ของถนนในทางแยกว่าต้องการเชื่อมต่อที่ด้านใดของทางแยก

4.1.6.4 คลาส EditorFrame

เป็น Frame ในการติดต่อผู้ใช้เพื่อใช้ในการสร้างวัตถุขึ้นมาในระบบสัญญาณไฟจราจร โดยจะมีปุ่มต่างๆ ให้ผู้ใช้เลือกว่าต้องการจะสร้างวัตถุชนิดใด เมื่อกดปุ่มแล้วก็คลิกที่แผนที่ (MapFrame) เพื่อใช้ในการสร้างวัตถุขึ้นมาในแผนที่ จะมีลักษณะของปุ่มต่างๆ แสดงได้ดังรูป

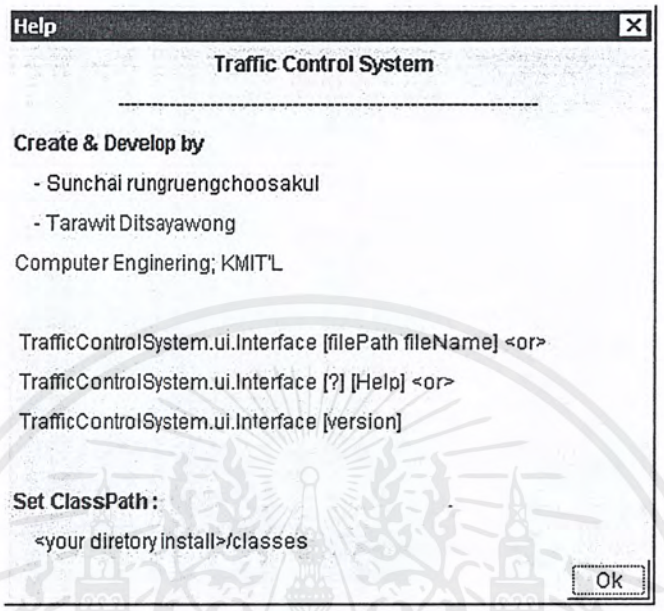


รูปที่ 4-8 แสดงลักษณะหน้าจอของ EditorFrame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6.5 คลาส HelpDialog

เป็นคลาสที่ใช้สำหรับแสดงผลการช่วยเหลือของโปรแกรม โดยจะมีในส่วนของการติดตั้งโปรแกรม และส่วนของการเรียกใช้งานโปรแกรม ดังรูป



รูปที่ 4-9 แสดงลักษณะหน้าจอของ HelpDialog

การเรียกใช้นั้นสามารถเรียกใช้ได้โดยตรง หรือผ่านทางคลาส Interface ก็ได้ (รายละเอียดดูได้จากหัวข้อถัดไป)

4.1.6.6 คลาส Interface

เป็นคลาสหลักในการทำงานของโปรแกรม โดยจะเก็บในส่วนของ Controller ในการสร้างและแก้ไขแผนที่เอาไว้ ซึ่งเป็นส่วนสำคัญในการทำงานของโปรแกรม (ส่วน Controller ของการจำลองการทำงานของระบบสัญญาณไฟจราจรนั้น จะถูกส่งมาให้จาก Controller ของคลาส Interface อีกทอดหนึ่ง) ซึ่งคลาสนี้เป็นคลาสหลักในการทำงานของโปรแกรม การทำงานทุกอย่างของโปรแกรมจะต้องผ่านที่คลาสนี้ไม่ว่าจะเป็นการสร้างแผนที่ การเปิดแผนที่ การบันทึกแผนที่ หรือว่าจะเป็นการจำลองการทำงานของระบบสัญญาณไฟจราจรเองก็ตามก็ต้องผ่านที่คลาสนี้

การสั่งเริ่มต้นสั่งงานโปรแกรมนั้น สามารถสั่งงานได้ดังนี้

"java TrafficControlSystem.ui.Interface [fileName filePath] [?] [Help] [version]"

โดยที่

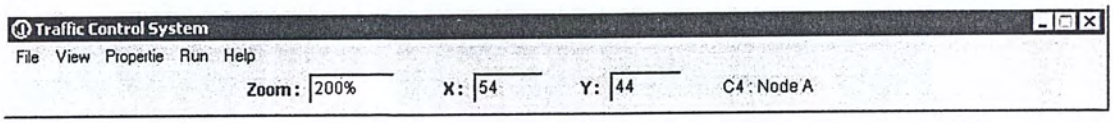
fileName filePath เป็นชื่อของไฟล์ และโฟลเดอร์ของไฟล์ที่ต้องการเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

? เป็นการแสดงหน้าจอของการช่วยเหลือการทำงาน

Help เป็นการแสดงหน้าจอของการช่วยเหลือการทำงานเช่นเดียวกับ ?

version เป็นการแสดงเวอร์ชันของโปรแกรมว่าเป็นเวอร์ชันใด

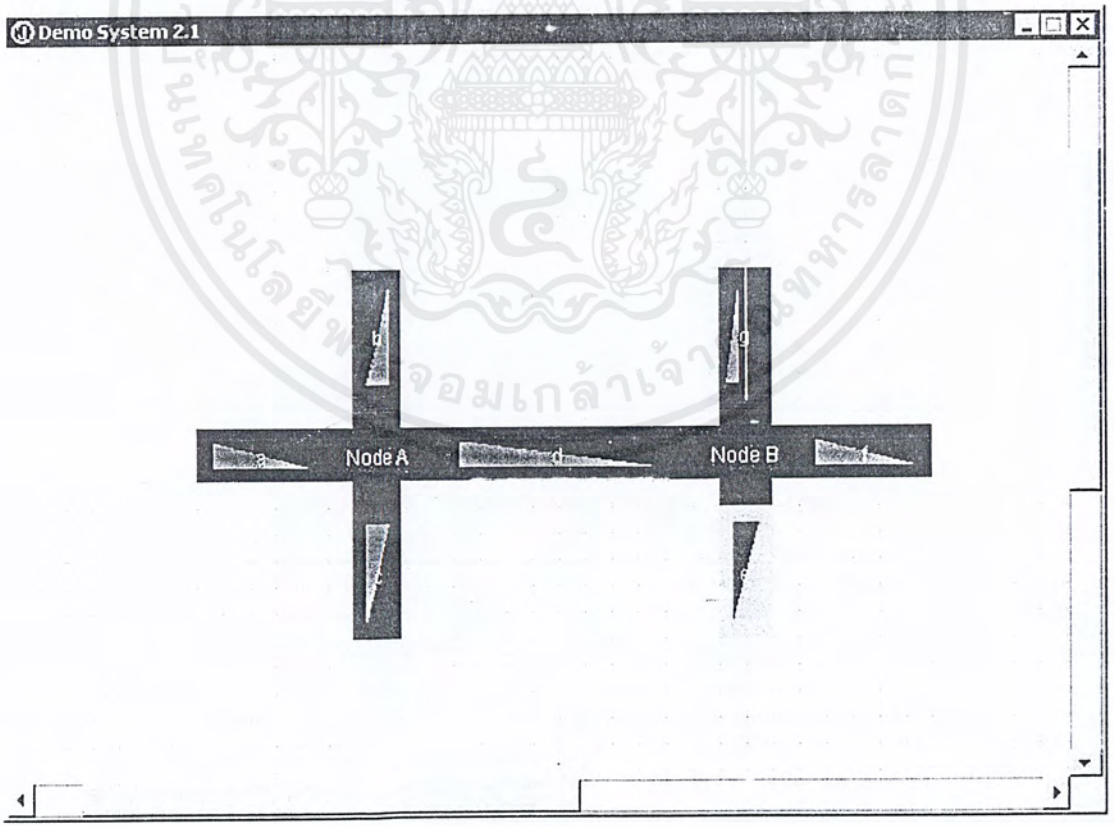


รูปที่ 4-10 แสดงลักษณะหน้าจอของ Interface

4.1.6.7 คลาส MapFrame

เป็นคลาสที่ใช้ในการแสดงผลของแผนที่ของระบบ ทั้งการแสดงผลในส่วนของการสร้างและแก้ไขแผนที่ และในส่วนของการจำลองระบบสัญญาณไฟจราจรด้วย ลักษณะของหน้าจอเป็นดังรูปที่ 4-11

จากรูปจะเห็นว่ามีส่วนของ Scroll Bar ใช้ในการเลื่อนแผนที่ ในกรณีแผนที่มีขนาดใหญ่ และจะมีส่วนของ Mouse Motion Listener ด้วย (ใช้ในการตรวจจับเหตุการณ์ของเมาส์ว่ามีการทำงานอยู่ในสถานะใด ซึ่งการทำงานในส่วนนี้จะใช้คลาส MouseMotionListen ในการตรวจจับเหตุการณ์ดังกล่าว) ส่วนของ Title ของหน้าจอจะเป็นการแสดงชื่อของแผนที่ที่ทำงานอยู่



รูปที่ 4-11 แสดงหน้าจอของ MapFrame

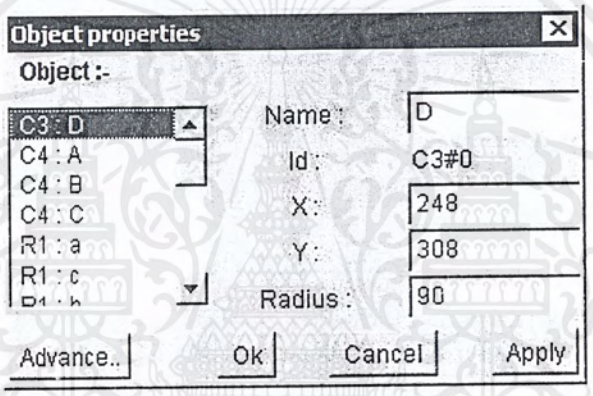
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6.8 คลาส MouseMotionListen

เป็นคลาสที่ใช้ในการตรวจจับเหตุการณ์ของเมาส์ โดยจะมีตรวจจับว่า Drag วัตถุหรือไม่ ถ้ามีการ Drag วัตถุก็จะทำการเลื่อนตำแหน่งของวัตถุไปยังที่ตำแหน่งของเมาส์นั้นๆ และใช้ในการแสดงผลตำแหน่งของเมาส์ที่อยู่ว่าอยู่ที่ตำแหน่งใดของแผนที่ โดยจะแสดงตำแหน่งที่ Text Field ของคลาส Interface (ตำแหน่งของแกน X และแกน Y จะสลับตำแหน่งกันเนื่องมาจากการแสดงผลของระบบ ดูรายละเอียดของการแสดงผลได้ที่หัวข้อ 3.2.3.13 คลาส Cal)

4.1.6.9 คลาส ObjectDialog

เป็นคลาสที่ใช้ในการแสดงผลวัตถุที่อยู่ในโปรแกรมทั้งหมดว่ามีวัตถุใดอยู่บ้าง และใช้ในการแก้ไขรายละเอียดของวัตถุนั้นได้ด้วย โดยรายละเอียดที่สามารถแก้ไขได้นั้นจะเป็นรายละเอียดที่ง่ายๆ คือ ชื่อ, ตำแหน่งของแกน X, ตำแหน่งของแกน Y และองศาของวัตถุ ลักษณะของหน้าจอมีลักษณะดังรูปที่ 4-12

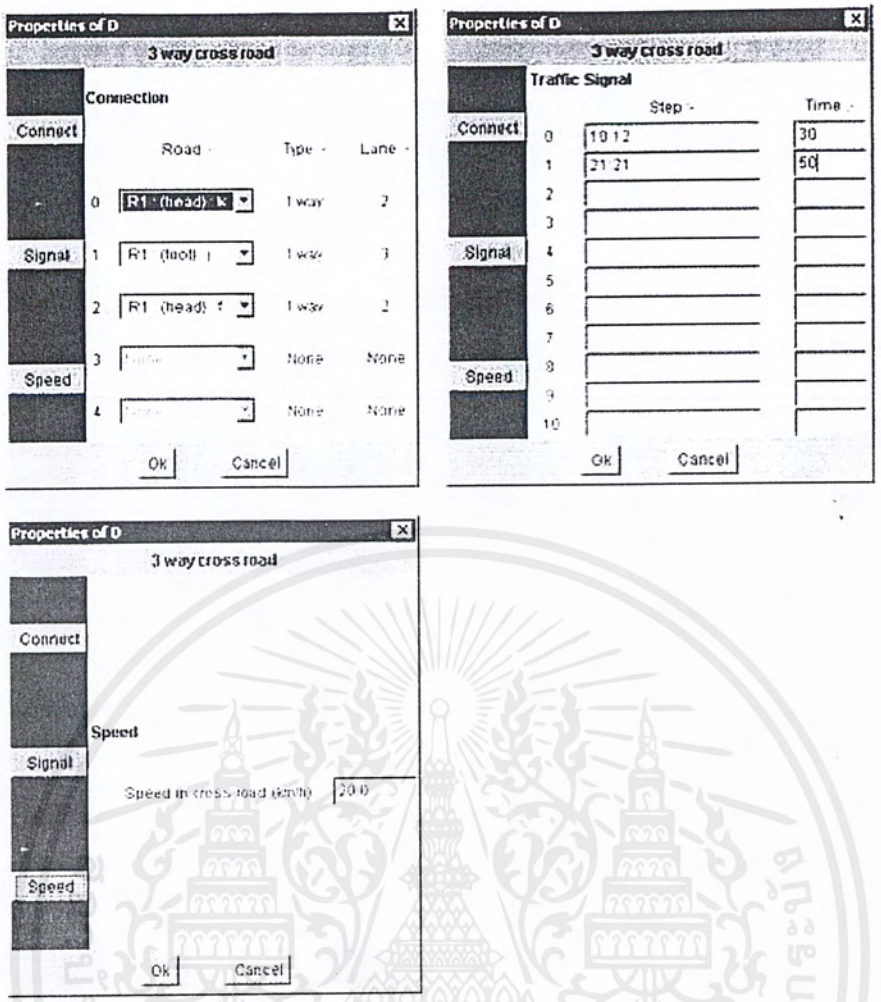


รูปที่ 4-12 แสดงหน้าจอของ ObjectDialog

4.1.6.10 คลาส PropertiesCrossDialog

เป็น Dialog ที่ใช้ในการแก้ไขค่าของทางแยก ไม่ว่าจะเป็นสามแยก, สี่แยก หรือห้าแยกในระบบ โดยจะมีลักษณะของหน้าจอเป็นดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



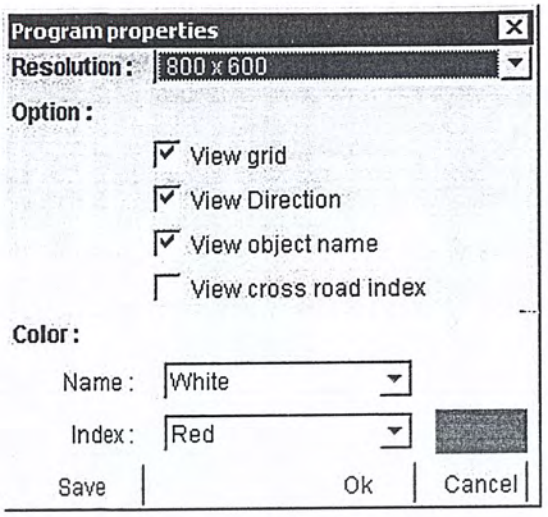
รูปที่ 4-13 แสดงหน้าจอของ CrossDialog

จากรูป จะเห็นได้ว่ามีส่วนของการแก้ไขอยู่ 3 ส่วนคือ ส่วนของการเชื่อมต่อกับถนน ส่วนของจังหวะของสัญญาณไฟจราจรของทางแยก และส่วนของความเร็วที่วิ่งได้บนทางแยกนั้นๆ (รายละเอียดของค่าแต่ละค่านั้นแสดงได้ในหัวข้อที่ 3.2.3.2 คลาสทางแยก)

4.1.6.11 คลาส PropertiesProgramDialog

เป็นคลาสที่ใช้แก้ไขรายละเอียดของโปรแกรม ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-14 แสดงลักษณะหน้าจอของ PropertiesProgramDialog

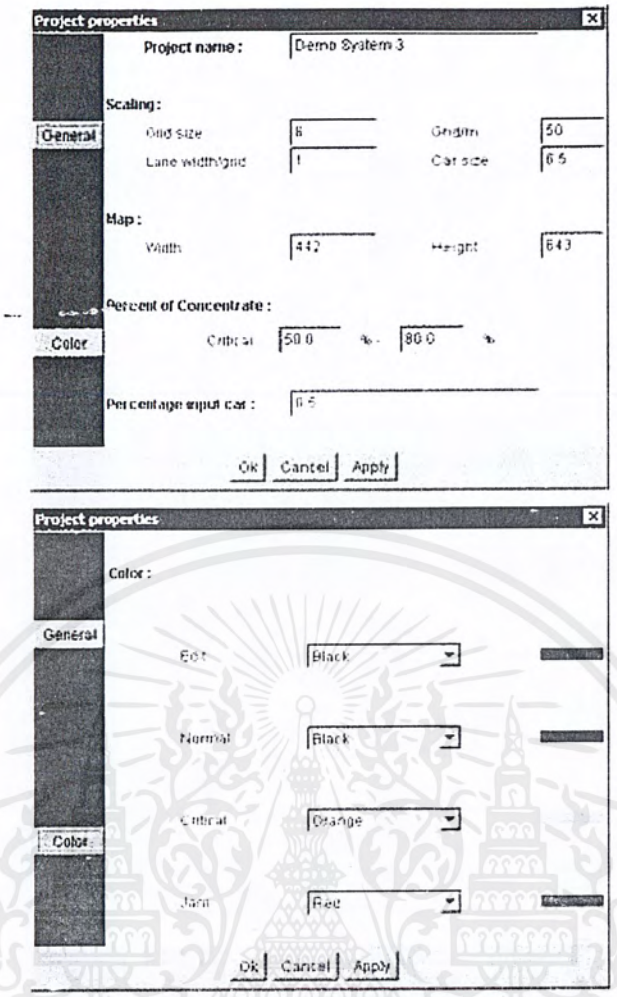
จากรูปจะเห็นได้ว่ามีส่วนของขนาดของหน้าจอในการแสดงผล, การแสดงผลของ Grid, การแสดงผลของทิศทางของถนน, การแสดงผลชื่อของวัตถุ, การแสดงผลของ index ของถนนที่ทางแยก, สีของตัวอักษรของชื่อที่แสดงผล และสีของการแสดงผลของ index ของถนนที่ทางแยก ซึ่งค่าต่างๆ ที่ได้กล่าวมาแล้วนั้นสามารถบันทึกลงไปในไฟล์ได้โดยจะได้ไฟล์ที่ชื่อ Config.config ซึ่งจะอยู่ในโฟลเดอร์เดียวกับที่ได้เรียกใช้งานโปรแกรม โดยจะถูกเปิดไฟล์ขึ้นมาใช้งานอย่างอัตโนมัติทันทีที่โปรแกรมถูกเรียกใช้งาน

4.1.6.12 คลาส PropertiesProjectDiaiog

ใช้งานสำหรับการแก้ไขรายละเอียดของคลาส Controller ทั้งหมด ซึ่งค่าที่สามารถแก้ไขได้นั้นจะมีดังนี้

ชื่อของแผนที่, ขนาดของ Grid ที่ใช้แสดงผล, ขนาด Grid ต่อเลนของถนน, ขนาดของความยาวของถนนต่อ 1 Grid, ขนาดความยาวของรถ, ความกว้างของแผนที่, ความสูงของแผนที่, เปอร์เซ็นต์ของปริมาณรถที่เข้ามายังถนนจากถนนปลายเปิด, เปอร์เซ็นต์ต่ำสุดของปริมาณรถที่อยู่ในถนนแล้วถือว่ารถไม่ติด (Low Critical), เปอร์เซ็นต์สูงสุดของปริมาณรถที่อยู่ในถนนแล้วถือว่าสูงติดแบบ Critical (High Critical) และสีของถนนในการเปรียบเทียบแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-15 แสดงลักษณะหน้าจอของ PropertiesProjectDialog

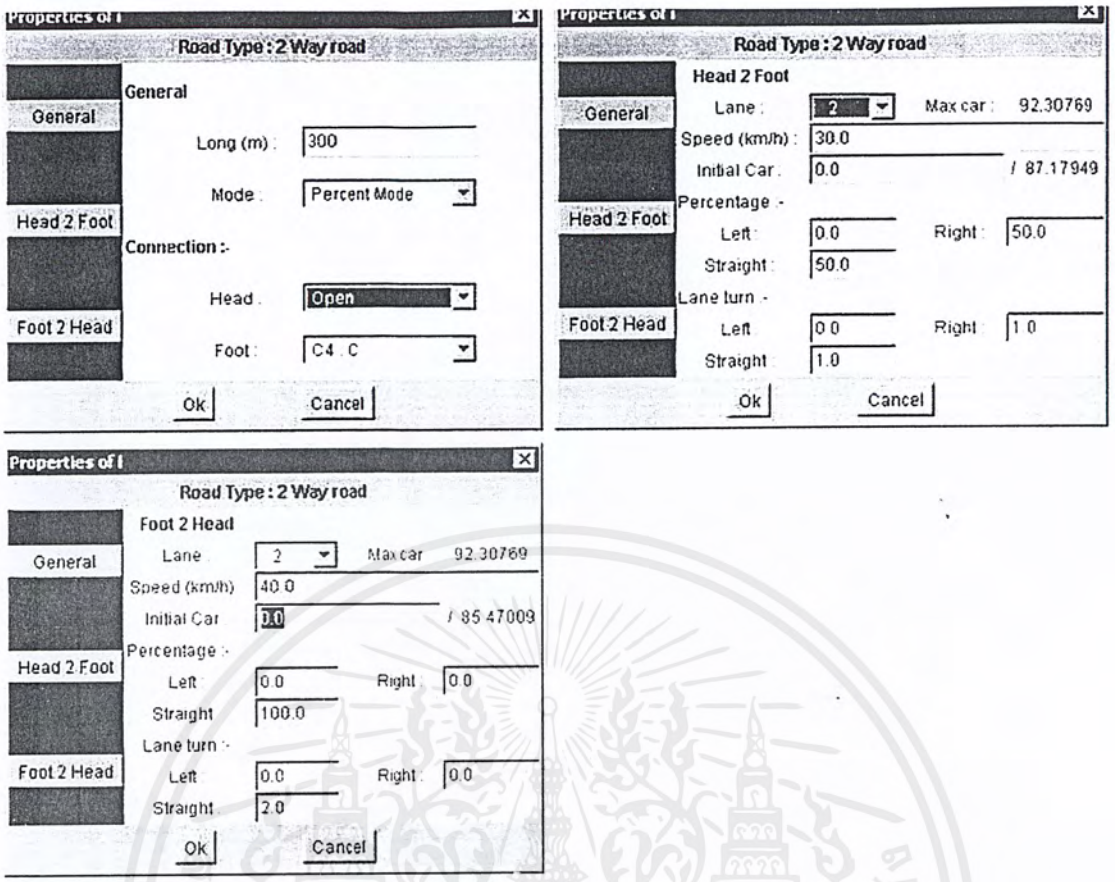
4.1.6.13 คลาส PropertiesRoadDialog

เป็น Dialog ในการแก้ไขรายละเอียดของถนนที่อยู่ในระบบ ทั้งถนนทางเดียวและถนนสองทาง ซึ่งจะมีหน้าจอของการทำงานเป็นดังรูปที่ 4-16

ซึ่งจาก Dialog จะสามารถแก้ไขรายละเอียดของถนนได้ โดยมีค่าที่แก้ไขได้คือ ค่าความกว้างของถนน, ความยาวของถนน, ปริมาณรถเริ่มต้นในถนน, เปอร์เซ็นต์ของการเลี้ยวของรถในถนน และจำนวนเลนที่ให้รถเลี้ยวว่ามีกี่เลน (รายละเอียดของค่าต่างๆ ดังกล่าวนี้ดูได้ที่หัวข้อ 3.2.3.6 – 3.2.3.8 คลาสถนน คลาสถนนทางเดียว และคลาสของถนนสองทาง)

จาก Dialog จะแบ่งหน้าจอของการแก้ไขออกเป็น 3 ส่วน คือ ส่วนของการเชื่อมต่อ ส่วนของถนนด้าน Head ไปยังด้าน Foot และส่วนของถนนจากด้าน Foot ไปยังด้าน Head (รายละเอียดของถนนสามารถดูได้ที่หัวข้อ 3.2.3.6 คลาสถนน)

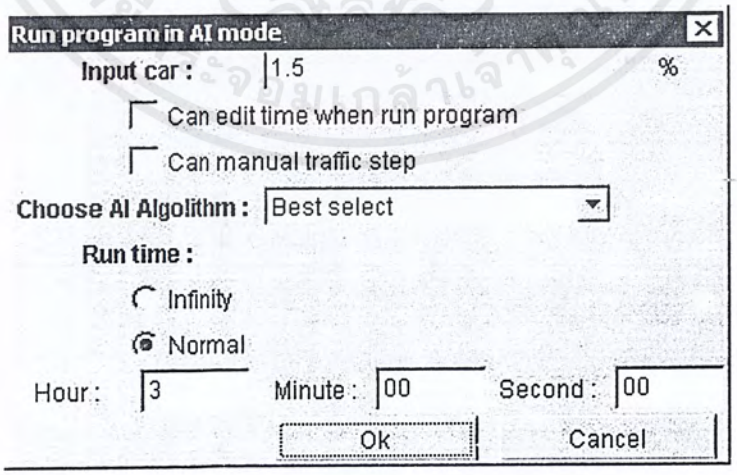
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-16 แสดงลักษณะของหน้าของ PropertiesRoadDialog

4.1.6.14 คลาส RunAIDialog

เป็น Dialog ที่ใช้ในการกำหนดค่าต่างๆ ของการจำลองการทำงานของสัญญาณไฟจราจรแบบ AI ซึ่งจะมีหน้าจอของการทำงานเป็นดังรูป



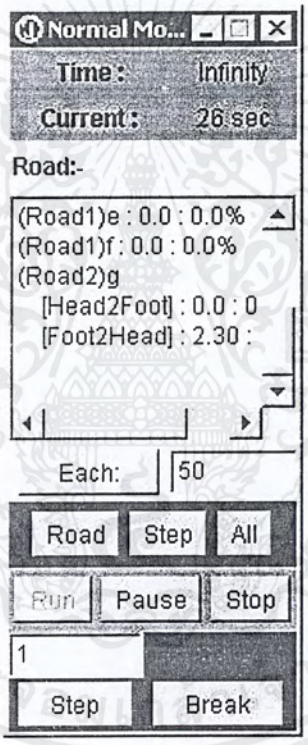
รูปที่ 4-17 แสดงหน้าจอของ RunAIDialog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้สามารถเลือกอัลกอริทึมในการทำงานของระบบสัญญาณไฟจราจรได้ และยังสามารถกำหนดสัญญาณไฟจราจรเอง หรือปรับเวลาในการปล่อยสัญญาณไฟจราจรได้ด้วย (ในการใช้งานจริงนั้นต้องสามารถทำงานทั้งสองอย่างไปพร้อมกันได้ด้วย เพราะว่าอาจจะมีคนคอยควบคุมการทำงานที่ทางแยกอีกทีหนึ่งด้วยก็ได้ ไม่จำเป็นจะต้องมีการทำงานเฉพาะในส่วนของคอมพิวเตอร์เพียงอย่างเดียว)

4.1.6.15 คลาส RunningFrame

เป็นส่วนที่สำคัญมากในการจำลองการทำงานของระบบสัญญาณไฟจราจร เพราะภายในวัตถุนี้จะมีวัตถุของคลาส RunThread อยู่ภายในด้วย โดยจะมี RunThread เป็น Thread ในการจำลองระบบสัญญาณไฟจราจร และจะมี Thread ที่ใช้ในการแสดงผลคอยทำการแสดงผลของระบบที่ทำงานได้ว่าเป็นอย่างไร ลักษณะของหน้าจอจะมีลักษณะดังรูป



รูปที่ 4-18 แสดงหน้าจอของ RunningFrame

จากรูปจะมีปุ่มคอยควบคุมการทำงานของการจำลองระบบสัญญาณไฟจราจรอยู่ และมีปุ่มคอยควบคุมการทำงานของ Monitoring อยู่ด้วย โดยปุ่มในแถบสีเทานั้นจะเป็นปุ่มที่คอยควบคุมการทำงานของ Thread การจำลองระบบสัญญาณไฟจราจร ส่วนปุ่มในแถบสีน้ำเงินนั้นจะเป็นปุ่มที่คอยควบคุมการทำงานในส่วนของการ Debug การทำงานของการจำลองระบบสัญญาณไฟจราจรซึ่งสามารถตั้งจุดหยุดได้ และสามารถ Step ขั้นตอนการทำงานของสัญญาณไฟจราจรอีกด้วย ส่วนปุ่มในแถบสีดำด้านบนนั้นจะเป็นปุ่ม

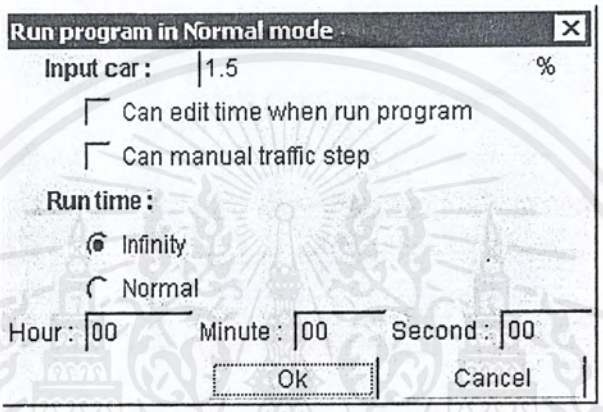
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่คอยดูค่าที่วัดได้จากการ Monitoring โดยจะแสดงผลอยู่ในรูปของ Text Field อยู่ตลอดเวลา (รายละเอียดของ Monitoring นั้นดูได้จากหัวข้อ 4.6.1 คลาส Monitoring)

ในส่วนของ Title ของ Frame จะแสดงว่าจำลองระบบสัญญาณไฟจราจรอยู่ในรูปแบบใด จากรูปนั้นเป็นการทำงานในโหมดของสัญญาณไฟจราจรแบบปรกติ (Normal Mode)

4.1.6.16 คลาส RunNormalDialog

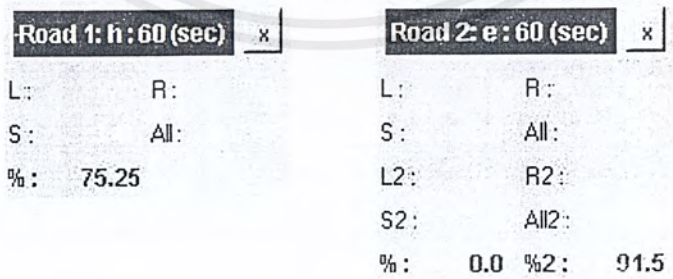
เป็น Dialog ในการใส่ค่าต่างๆ เพื่อใช้ในการจำลองการทำงานของสัญญาณไฟจราจรในโหมดการทำงานแบบปรกติ (มีการตั้งจังหวะของสัญญาณไฟจราจรตามเวลา) มีลักษณะของหน้าจอดังรูป



รูปที่ 4-19 แสดงหน้าจอของ RunNormalDialog

4.1.6.17 คลาส ShowRoad

เป็น Windows ที่ใช้ในการแสดงผลของถนน ทั้งถนนทางเดียวและถนนสองทาง โดยจะแสดงผลทั้งรายละเอียดของรถที่อยู่ในถนน (รถที่กำลังอยู่ใน Queue ของถนน), เปอร์เซ็นต์ของรถที่อยู่ในถนน และรถที่กำลังจะเลี้ยวด้วย โดยจะมีลักษณะหน้าจอเป็นดังรูป



รูปที่ 4-20 แสดงหน้าจอของ ShowRoad

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นได้ว่ามีการแสดงผลของถนนสองแบบ คือ ถนนทางเดียว (ด้านซ้าย) และถนนสองทาง (ด้านขวา) โดยมีปุ่ม "X" เพื่อใช้ในการปิดการแสดงผล

4.1.6.18 คลาส ShowTrafficStep

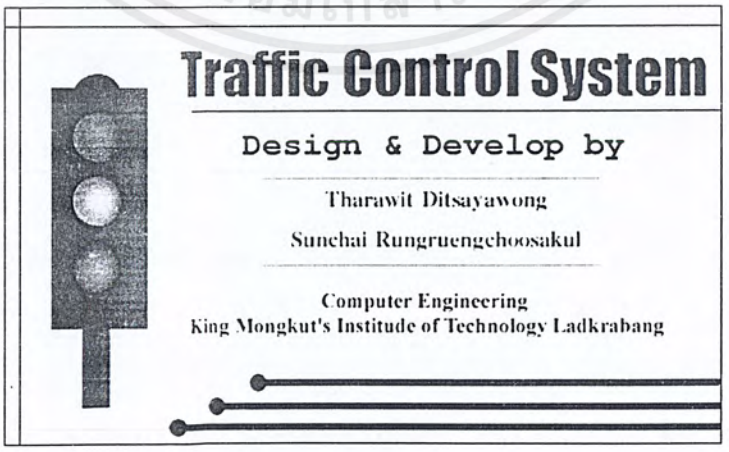
เป็น Windows ที่ใช้ในการแสดงผลของทางแยก โดยจะมีการแสดงจังหวะของสัญญาณไฟจราจรเอาไว้ (เป็นตัวอักษรหนา) ถ้ามีการกำหนดว่าผู้ใช้สามารถตั้งสัญญาณไฟจราจรเองได้นั้นก็จะมี Radio Button เพื่อให้ผู้ใช้สามารถเลือกจังหวะของสัญญาณไฟจราจรได้ ลักษณะของหน้าจอการแสดงผลนั้นมีลักษณะดังรูป

รูปที่ 4-21 แสดงหน้าจอของ ShowTrafficStep

จากรูป ด้านซ้ายจะเป็นการแสดงผลของทางแยก B ที่เวลา 60 วินาที โดยที่ผู้ใช้ไม่สามารถกำหนดจังหวะของสัญญาณไฟจราจรได้ ส่วนด้านขวาจะเป็นหน้าจอของการแสดงผลที่ผู้ใช้สามารถกำหนดจังหวะของสัญญาณไฟจราจรได้ และยังสามารถกำหนดเวลาเองได้ด้วย

4.1.6.19 คลาส SplashScreen

เป็น Splash Screen ของโปรแกรม โดยจะถูกเรียกใช้เมื่อตอนเริ่มการใช้งานโปรแกรม จะเป็น Windows ของการทำงานโดยทำการโหลดภาพ SplashScreen.jpg ขึ้นมาแสดงผล ดังรูป



รูปที่ 4-22 แสดงหน้าจอของ SplashScreen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6.20 Sub package io

เป็น Package ที่ใช้ในการทำงานที่เกี่ยวข้องกับอุปกรณ์ Input Output ของโปรแกรมทั้งหมด ซึ่งภายใน Package จะประกอบด้วยคลาสเพียง 1 คลาส คือ คลาส SaveOutput

4.1.6.20.1 คลาส SaveOutput

เป็นคลาสที่ใช้ในการบันทึกข้อมูลจากการ Monitoring ได้ลงไปเป็นไฟล์ เพื่อใช้ในการประเมินผลการทำงานของสัญญาณไฟจราจรที่ได้คิดขึ้น การทำงานของคลาสนี้จะถูกเรียกใช้เมื่อมีการจำลองการทำงานเสร็จสิ้นแล้ว

ไฟล์ที่ได้จากการบันทึกจะเป็นไฟล์ชนิด Text โดยจะอยู่ใน Format ของ UNIX (เพราะว่าภาษา JAVA แปลงไฟล์เป็นไฟล์ชนิดนั้นเอง)

4.2 ปัญหาและการแก้ไข

ในการพัฒนาระบบ จะต้องมีข้อผิดพลาด (หรืออาจจะเรียกได้ว่าเป็น Bug) ของการทำงานเป็นเรื่องธรรมดา แต่ว่าเมื่อเกิดข้อผิดพลาดแล้วจำเป็นต้องแก้ไขให้ระบบทำงานได้อย่างถูกต้อง ข้อผิดพลาดของระบบนั้นอาจจะเกิดมาจากผู้พัฒนาเอง หรืออาจจะเกิดมาจากเครื่องคอมพิวเตอร์เองก็ได้ ข้อผิดพลาดของระบบมีดังนี้

ข้อผิดพลาด	การแก้ไข
การออกแบบปริมาณรถในถนนเป็นจำนวนจริง	การนิยามปริมาณรถเป็นจำนวนจริงนั้น เพื่อให้สามารถทำงานในหน่วยของ Car/(sec*lane) ได้ ซึ่งจะเกิดปัญหาที่ทางแยก เนื่องจากว่าในความเป็นจริงนั้นจะมีการเคลื่อนที่ของรถเป็นจำนวนเต็มเท่านั้น จึงมีค่าของ car[] ในทางแยกเพื่อเก็บค่าของปริมาณรถที่วิ่งได้เป็นจำนวนเต็ม โดยจะทำการปรับค่าของรถในวินาทีสุดท้ายเท่านั้น (ดังนั้นปริมาณรถที่วิ่งได้ในระหว่างทางแยกนั้นจะมีข้อผิดพลาดประมาณ ±1 คัน/ 1 จังหวะสัญญาณไฟจราจร
การเกิด Dead Lock	ในส่วนของการจำลองระบบสัญญาณไฟจราจรนั้น บางทีอาจจะเกิดปัญหาในส่วนของ Thread การแสดงผลกับ Thread ของการจำลองระบบสัญญาณไฟจราจรเกิดการรอกัน ซึ่งแก้ไขได้โดยการจัดการอ้างอิงการใช้ตัวแปรอย่างถูกต้องและมีลำดับการทำงานที่ถูกต้อง เพื่อให้ทำงานได้มีประสิทธิภาพและไม่เกิด Dead Lock
การแสดงผลของวัตถุ	ในการแสดงผลของวัตถุนั้นมีการอาศัยการสร้างรูปอย่างง่ายๆ และใช้การแสดงผลเป็นรูปของวัตถุ แต่จะเกิดปัญหาในการย่อหรือขยายแผนที่ จึงอาศัยการทำงานของ Computer Graphic ในการคำนวณการแสดงผล โดยจะเป็นการแสดงผลใน 2 มิติแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริมาณรถที่เข้าสู่ระบบ	ปริมาณรถที่เข้าและออกจากระบบ ในทางปฏิบัตินั้นมีปัจจัยของช่วงเวลาเข้ามาเกี่ยวข้องด้วย เช่นในเวลาเร่งรีบเป็นต้น ในระบบจึงได้ออกแบบให้มีการหาปริมาณค่าของรถที่เข้าสู่ระบบ โดยการหาเป็นเปอร์เซ็นต์ของรถในถนนแทน
ความเร่งของรถ	เป็นปัญหาที่สำคัญที่สุดของระบบ เนื่องจากในทางปฏิบัติจริงนั้นรถทุกคันจะมีความเร่งเป็นของตนเอง (หมายความว่าจะมีความหน่วงของเวลาในการวิ่งของรถด้วย ว่ามีความหน่วงเท่าใด) แต่ในระบบที่จำลองการทำงานขึ้นไม่มีการคิดในส่วนนี้ โดยจะออกแบบให้ลักษณะของการวิ่งของรถนั้นมีลักษณะของการวิ่งเป็นแบบ Simple คือวิ่งด้วยอัตราเร็วคงที่ และไม่มีความเร่ง
การกำหนดจังหวะของสัญญาณไฟจราจร	การกำหนดสัญญาณไฟจราจรนั้น เป็นส่วนสำคัญ เพราะว่าการกำหนดสัญญาณไฟจราจรที่ผิดนั้นจะทำให้ระบบทำงานไม่ถูกต้อง ดังนั้นการกำหนดสัญญาณไฟจราจรให้ถูกต้องนั้นจำเป็นจะต้องมีการตรวจสอบ ซึ่งในระบบที่ได้ออกแบบไว้ได้มีการตรวจสอบสัญญาณไฟจราจรแล้วว่ามีการทำงานที่ถูกต้องเท่านั้น จึงจะสามารถทำงานได้
การกำหนดสัญญาณไฟจราจรในกรณีที่มีจังหวะสัญญาณไฟจราจรเดียว	การกำหนดสัญญาณไฟจราจรในกรณีที่มีสัญญาณไฟจราจรเดียวนั้นเป็นส่วนที่สำคัญเพราะว่าในทางปฏิบัติจริงนั้นมีทางแยกประเภทนี้อยู่ การทำงานของทางแยกปะเภทนี้จะเป็นการทำงาน โดยการกำหนดจังหวะของสัญญาณไฟจราจรลงไปเลย แล้วกำหนดเวลาของการทำงานของสัญญาณไฟจราจรนั้นเป็น –1 เพื่อให้ทางแยกนั้นปล่อยสัญญาณไฟจราจรนั้นไม่มีที่สิ้นสุด
การวิ่งของรถในทางแยกจะไม่เท่ากันในกรณีที่รถเลี้ยวกับวิ่งตรงไป	ลักษณะของการวิ่งของรถในทางแยกนั้นเป็นสิ่งที่สำคัญมากเนื่องจากว่าปัจจัยในการวิ่งของรถในทางแยกนั้นมีสาเหตุมาจาก เส้นทางการวิ่งของรถ จำนวนเลนการวิ่งของรถที่เลี้ยว จำนวนเลนของถนนที่รับ ปริมาณรถในถนนที่รับ ฯลฯ ซึ่งปัจจัยเหล่านี้เป็นปัจจัยที่มีผลต่อการวิ่งของรถในทางแยกทั้งสิ้น ดังนั้นในกรณีที่รถที่เลี้ยวในทางแยก กับส่วนของรถที่วิ่งตรงไปนั้น ถือว่ามีค่าใกล้เคียงกัน เพื่อความง่ายต่อการคำนวณ
การกำหนดสัญญาณไฟจราจรที่ทางแยกในกรณีที่มีการตั้งสัญญาณไฟจราจรไม่พอดี	การกำหนดสัญญาณไฟจราจรของทางแยกในทางแยกที่ไม่พอดีกัน คือมีการเหลื่อมล้ำกันในจังหวะของสัญญาณไฟจราจรนั้น ในระบบที่ได้ออกแบบไว้ไม่ได้มีการออกแบบในส่วนดังกล่าวเอาไว้ แต่ว่าถ้าเรากำหนดสัญญาณไฟจราจรอย่างดีแล้วเราจะน่าจะได้จังหวะของสัญญาณไฟจราจรที่ดีที่สุดในการทำงานโดยมีการแบ่งจังหวะของการทำงานของสัญญาณไฟจราจรอย่างแน่นอน
การวิ่งของรถในถนนในกรณีที่มีการจอดรถในถนน	ในกรณีของถนนที่มีรถจอดอยู่ในถนนนั้น จะเป็นถนนที่มีการทำงานที่เหมือนกับการที่ถนนนั้นสูญเสียพื้นผิวการจราจรไป (จำนวนเลนของถนนนั้นลดลงนั่นเอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถนนที่มีตรอกและซอย	การที่ถนนนั้นมีตรอกและซอยนั้น เราจะต้องพิจารณาถนนที่มีตรอกละซอยนั้นเป็นถนนที่มีปริมาณรถเข้าสู่ถนนมากหรือน้อย ถ้ามีน้อยก็ไม่ควรจะนำมาคิด แต่ว่าถ้ามีมากก็แสดงว่าถนนนั้นน่าจะมีการจัดระบบสัญญาณไฟจราจรอยู่แล้ว แล้วถ้าเกิดว่าตรอกและซอยเหล่านั้นไม่มีการจัดระบบสัญญาณไฟจราจรอยู่ก็จะมองว่าเป็นทางแยกที่มีการตั้งสัญญาณไฟจราจรแบบจังหวะเดียวก็ได้
ถนนสั้นมาก	ในกรณีที่ถนนสั้นมาก ในระบบที่ได้ออกแบบไว้จะมองว่าถนนนั้นไม่มี Queue อยู่ ซึ่งก็หมายความว่ารถที่เข้าสู่ถนนนั้นจะไม่มีการวิ่งในถนนนั้นนานมาก เพราะว่าเป็นการวิ่งเข้าไปยัง Buffer ที่รอการคำนวณทิศทางของการเลี้ยวซ้าย ขวาของถนนเลย ซึ่งก็ใกล้เคียงกับความเป็นจริง
รถวิ่งคร่อมเลน	การที่รถวิ่งคร่อมเลนนั้น ถือว่าเป็นการวิ่งของรถที่จัดการได้ลำบาก เพราะว่านอกจากจะทำให้รถวิ่งได้น้อยลงแล้ว ยังทำให้การทำงานในระบบผิดพลาดได้ โดยในระบบที่ได้ออกแบบไว้ไม่ได้มีการคิดในส่วนนี้ และการทำงานในส่วนนี้
ระบบทำงานช้า	มีสาเหตุมาจาก JAVA เองและก็เครื่องของผู้ใช้เอง ไม่เกี่ยวกับการทำงานของระบบ โดยโหมดที่จะทำงานช้านั้นจะเป็นโหมดของการทำงานแบบ AI ทั้งหมด เพราะว่ามีการคำนวณสัญญาณไฟจราจร
ปัญหาของการใช้ JVIEW ของWindows ในการทำงาน	เนื่องจาก JVIEW ของ Window มีการรองรับการทำงานของ JAVA ด้วย แต่ว่าคลาสที่ JVIEW ใช้นั้นเป็นคลาสที่มีการทำงานที่ช้ากว่าของ Sun ดังนั้นการทำงานในบาง Method นั้นใน JVIEW เองจะยังไม่รู้จัก Method เหล่านั้น จึงเป็นสาเหตุที่ไม่สามารถทำงานได้ ในการแก้ไขนั้นจำเป็นจะต้องใช้ Java Runtime Environment ในการทำงาน ซึ่งในระบบที่พัฒนาขึ้นนั้นอาศัย Java Development Kit เวอร์ชัน 1.3 ในการพัฒนา
ตัวอักษรที่ใช้แสดงผลมีขนาดเล็ก	สาเหตุเกิดจากจาก Java Runtime Environment เอง โดยสามารถแก้ไขได้โดยการลงไฟล์ font.properties.th ลงไปแทนที่ (ในกรณีที่ใช้ระบบปฏิบัติการแบบ Windows) หรืออาจจะเปลี่ยน Region ของเครื่องไปเป็นแถบยุโรปหรืออเมริกาแทนก็ได้
การแสดงผลของสัญญาณไฟจราจรค้าง หรือเวลาที่ใช้แสดงผลนั้นไม่ต่อเนื่อง	การแสดงผลของสัญญาณไฟจราจรในแผนที่นั้นค้าง เนื่องมาจากว่าการทำงานของการจำลองสัญญาณไฟจราจรนั้น ระบบได้ออกแบบการทำงานโดยออกแบบการทำงานออกเป็น Thread ย่อยๆ 2 Thread ด้วยกัน โดย Thread แรกนั้นจะเป็นส่วนของการจำลองระบบสัญญาณไฟจราจร และ Thread ที่สองนั้นเป็นส่วนของการแสดงผล (รวมทั้งการแสดงผลสัญญาณไฟจราจรบนแผนที่ด้วย) แต่การทำงานของ Thread ทั้งสองนั้นไม่เท่ากัน เพราะว่า Thread ที่ใช้ในการแสดงผลนั้นมีการทำงานที่ช้ากว่า เพราะมีการทำงานที่มากกว่า จึงเป็นสาเหตุให้การทำงานเกิดการกระตุก แก้ปัญหาโดยการใส่ค่าความหน่วงของเวลาเข้าไปใน Thread ของการจำลองระบบสัญญาณไฟจราจรเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนที่ที่สร้างทำงานไม่ถูกต้อง	เป็นสามเหตุมาจากตัวผู้ใช้เอง โดยระบบที่ออกแบบมานั้นจะให้ผู้ใช้สามารถตรวจสอบได้ว่าแผนที่ที่สร้างขึ้นนั้นถูกต้องหรือไม่ ถ้าแผนที่ไม่ถูกต้องระบบก็จะทำการฟ้องค่าที่ผิดพลาดขึ้นมา
สัญญาณไฟจราจรที่ทางแยกนั้นเข้าใจได้ยาก	เนื่องจากว่าสัญญาณไฟจราจรที่ทางแยกนั้นเป็นสิ่งสำคัญมาก ในกรณีที่บอกว่ามีการปล่อยสัญญาณไฟจราจรตรงไปนั้น ก็อาจจะมีการวิ่งของรถจากถนนหนึ่งไปยังอีกถนนหนึ่งได้หลายถนนก็ได้ (ในกรณีของห้าแยกเป็นต้น) จึงจำเป็นจะต้องกำหนดสัญญาณไฟจราจรขึ้นมาเพื่อให้เข้าใจได้ง่ายว่าสัญญาณไฟจราจรนั้นปล่อยรถในทิศทางใด ซึ่งในระบบได้แสดงผลด้านของถนนที่ทางแยกเอาไว้ด้วย เพื่อให้ผู้ใช้สามารถมองเห็นและเข้าใจตรงกันกับในระบบ
ปริมาณรถที่เข้าสู่ระบบและออกจากระบบมีไม่เท่ากัน	มีสาเหตุมาจากการปัดเศษของรถในถนนนั้นๆ มักจะพบในกรณีที่ระบบมีขนาดใหญ่ โดยสาเหตุนั้นอาจจะมาจากการปัดเศษของรถ หรืออาจจะเกิดจากการนำจำนวนจริงมาคูณกันแล้วค่าที่ได้ไม่ถูกต้องก็ได้ สำหรับระบบขนาดเล็กแล้วนั้นปัญหาดังกล่าวจะเกิดขึ้นน้อยมาก (ในการทดลองการทำงานปรากฏว่าค่าผิดพลาดที่เกิดขึ้นในระบบขนาดเล็กนั้นมีค่าเป็น 0.00002 คัน / ถนน * วินาทีเท่านั้น แต่สำหรับในระบบขนาดใหญ่ก็จะมีค่ามากกว่านี้

ตารางที่ 4-2 แสดงปัญหาและการแก้ไขของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

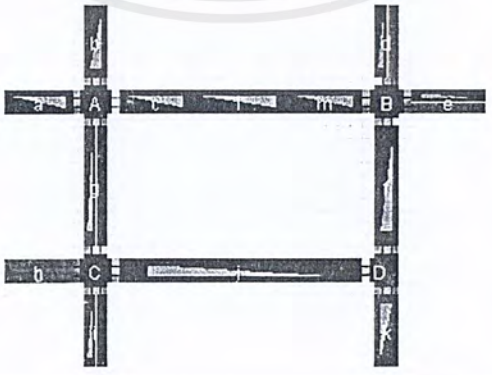
บทที่ 5
การทดสอบโครงงาน

การจำลองการทำงานของระบบสัญญาณไฟจราจรเป็นส่วนสำคัญที่ใช้ในการทดสอบการทำงานและอัลกอริทึมที่ทำการคิดและออกแบบเอาไว้ว่าถูกต้องหรือไม่อย่างไร การจำลองการทำงานของระบบสัญญาณไฟจราจรนั้นมีความสำคัญมาก เพราะเป็นการทดสอบและหาข้อผิดพลาดของโปรแกรมว่ามีส่วนใดผิดพลาดหรือไม่ และจะได้ทำการแก้ไขต่อไป อีกทั้งยังเป็นการประเมินผลการทำงานของอัลกอริทึมที่ได้ออกแบบมาว่าทำงานได้ประสิทธิภาพเป็นอย่างไรอีกด้วย

5.1 ผลลัพธ์ที่ได้จากการทดลอง

ผลลัพธ์ที่ได้จากโปรแกรมสามารถดูได้ 4 ส่วนคือ

1. จากส่วนของ MapFrame (ดูได้ในหัวข้อที่ 4.1.6.7) ซึ่งจะแสดงออกมาในระดับหนึ่งเป็นการมองภาพรวมของแผนที่ทั้งหมด
2. จากส่วนของ RunningFrame (ดูรายละเอียดได้ในหัวข้อที่ 4.1.6.15) จะแสดงรายละเอียดของค่าต่างๆ ที่ได้ Monitoring เอาไว้ โดยจะเป็นการดูการทำงานในขณะที่ทำการจำลองการทำงานของระบบสัญญาณไฟจราจรอยู่
3. จากส่วนของ ShowRoad และ ShowTrafficStep ซึ่งจะเป็นการดูรายละเอียดของถนนแบบเฉพาะเจาะจงว่ามีปริมาณรถในถนนเป็นเท่าใด หรือว่าจะเป็นสัญญาณไฟจราจรใดที่ทางแยกปล่อยอยู่ (รายละเอียดของ ShowRoad และ ShowTrafficStep สามารถดูได้ที่หัวข้อ 4.1.6.17 และ 4.1.6.18)
4. จากส่วนของ SaveOutput (ดูรายละเอียดได้ในหัวข้อที่ 4.1.6.20.1) แสดงผลลัพธ์ที่ได้จากการจำลองการทำงานของสัญญาณไฟจราจร ซึ่งจะเป็นการบันทึกค่าที่ได้จาก Monitoring มาประเมินผลการทำงานของสัญญาณไฟจราจรที่ได้



รูปที่ 5-1 แสดงลักษณะของหน้าจอจาก MapFrame

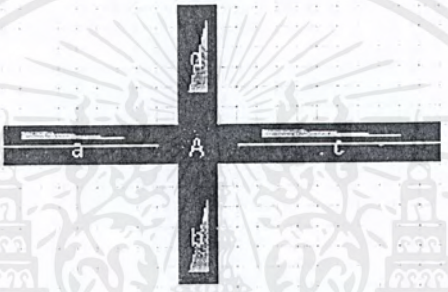
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะมีการ Update เฉพาะวัตถุที่มีการเปลี่ยนสีเท่านั้น เพื่อให้โปรแกรมมีเวลาในการคำนวณมากขึ้น ไม่เสียเวลาในการรอการแสดงผลของหน้าจอ

5.2 ผลการทดลองที่ได้

ผลการทดลองที่ได้นั้น มีความสำคัญมาก เพราะว่าจะเป็นการประเมินการทำงานของระบบสัญญาณไฟจราจรที่ได้ออกแบบมาว่ามีการทำงานเป็นอย่างไร มีประสิทธิภาพของการทำงานเป็นอย่างไร โดยจะใช้แผนที่ 3 แผนที่ในการทดสอบ โดยจะมีแผนที่ทั้ง 3 ขนาด ดังนี้

แผนที่ที่ 1 เป็นแผนที่การทำงานที่มีลักษณะดังรูป



รูปที่ 5-2 แสดงลักษณะของแผนที่ที่ 1 ที่ใช้ในการทดสอบ

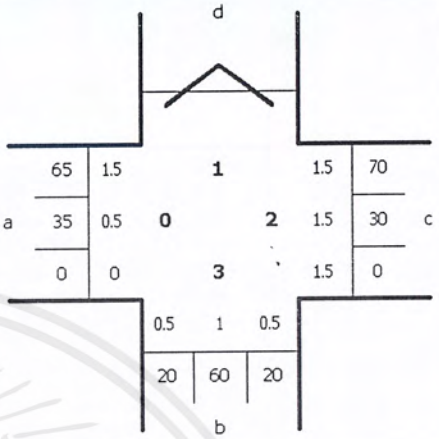
จากรูปจะเห็นว่าเป็นแผนที่การทำงานที่ง่ายที่สุด โดยจะมีทางแยก A เป็นสี่แยก และมีถนน a, b, c และ d ในแผนที่ รายละเอียดของถนนต่างๆ แสดงได้ดังตาราง

ชื่อของถนน	ความยาว (เมตร)	ด้าน Head ไปยังด้าน Foot			ด้าน Foot ไปยังด้าน Head		
		ความเร็ว (Km/h)	ความกว้าง (เลน)	ปริมาณรถเริ่มต้น	ความเร็ว (Km/h)	ความกว้าง (เลน)	ปริมาณรถเริ่มต้น
a	500	30	2	0	60	3	0
b	500	30	2	0	-	-	0
c	1000	45	3	0	100	5	0
d	500	30	2	0	-	-	0

ตารางที่ 5-1 แสดงรายละเอียดของถนนในแผนที่ที่ 1

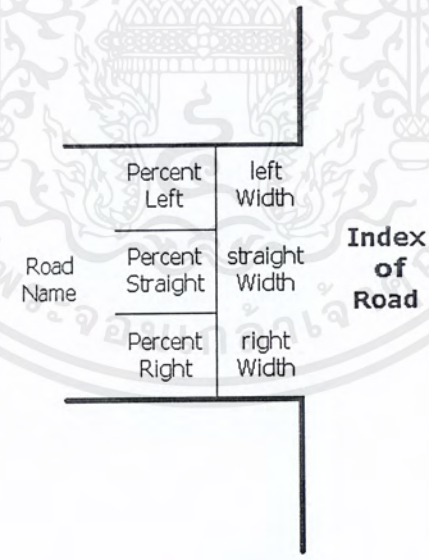
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของทางแยกมีการตั้งสัญญาณไฟจราจรเป็นดังตาราง

ชื่อของทางแยก	จังหวะของสัญญาณไฟจราจร	เวลา (วินาที)	ความเร็ว (Km/h)	รายละเอียด
A	"01:02" "21:32:20" "30:31:32"	30 30 40	28	

ตารางที่ 5-2 แสดงรายละเอียดของทางแยกในแผนที่ที่ 1

จากรูปที่แสดงในช่องของรายละเอียดของทางแยกนั้น จะแสดงรายละเอียดของค่าเป็นดังนี้



รูปที่ 5-3 แสดงรายละเอียดของสัญลักษณ์ที่ใช้ในรายละเอียดของทางแยก

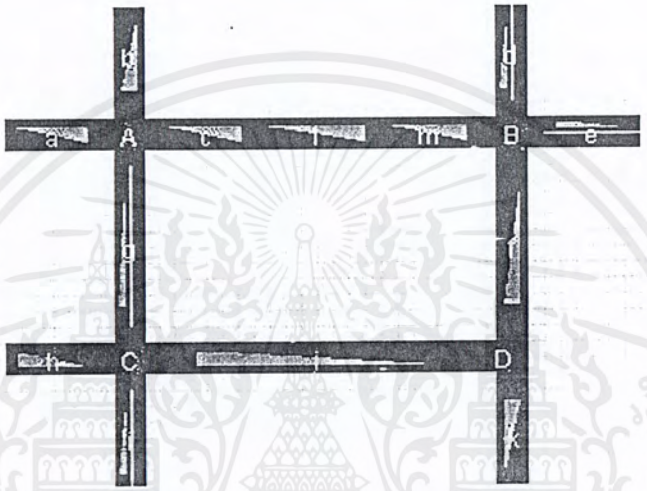
ในด้านที่แสดงมีค่านั้นจะเป็นด้านของถนนที่วิ่งเข้าสู่ทางแยก ส่วนถนนที่ไม่สามารถวิ่งเข้าสู่ทางแยกได้ (เป็นถนนทางเดียวที่รับการวิ่งของรถ หรือเชื่อมต่อกับทางแยกที่ด้าน Head) จะไม่มีการแสดงค่า โดยจะใช้เครื่องหมาย > แทนการวิ่งเข้าของรถ ส่วนในช่องของค่าที่เป็นไปไม่ได้ เช่นการวิ่งตรงไปของรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางตันข้างหน้า หรือเชื่อมต่อกับสามแยกที่ index ของถนนในทางแยกเป็น 1 เป็นต้น จะมีค่าเป็น – (Dash) แทนที่จะมีค่าเป็น 0 (เพราะว่าจะถือว่าค่า – เป็นค่าที่เป็นไปไม่ได้ ส่วนค่า 0 จะเป็นค่าที่ไม่มีการวิ่งของรถ แต่ในการสร้างแผนที่นั้นจะถือว่าค่า – และค่า 0 นั้นจะมีค่าเหมือนกัน คือ 0 ในการสร้างแผนที่)

แผนที่ที่ 1 นี้สามารถเปิดไฟล์ขึ้นมาทำงานได้โดยมีชื่อไฟล์ว่า Map_A.trf ซึ่งอยู่ในโปรแกรมที่โฟลเดอร์ Save/TestMap

แผนที่ที่ 2 เป็นแผนที่ที่มีขนาดกลาง โดยมีลักษณะของแผนที่ดังรูป



รูปที่ 5-4 แสดงลักษณะของแผนที่ที่ 2

จากรูปจะเห็นได้ว่ามีการเชื่อมต่อของถนน m, l และ c เป็นการเลียนแบบการเชื่อมต่อของถนนที่มีสะพานกั้นกลาง เนื่องจากว่ามีสะพานกั้นกลางจึงทำให้เกิดสภาพของ Bottle Neck หรือสภาพคอขวด โดยจะมีการรอของรถมากในกรณีที่มีรถเข้ายังถนนนั้นมาก แต่ว่าถ้ารถที่เข้ามาไม่มากก็จะไม่เกิดปัญหาดังกล่าว รายละเอียดของถนนในแผนที่แสดงได้ในตาราง

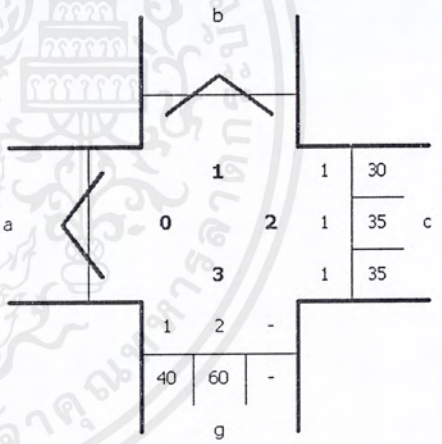
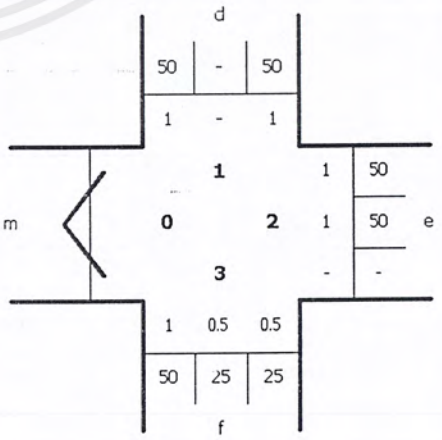
ชื่อของถนน	ความยาว (เมตร)	ด้าน Head ไปยังด้าน Foot			ด้าน Foot ไปยังด้าน Head		
		ความเร็ว (Km/h)	ความกว้าง (เลน)	ปริมาณรถเริ่มต้น	ความเร็ว (Km/h)	ความกว้าง (เลน)	ปริมาณรถเริ่มต้น
a	300	40	3	13	-	-	-
b	300	40	2	13	-	-	-
c	300	50	3	15	-	-	-
d	300	20	2	30	30	2	10
e	300	30	3	10	40	2	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

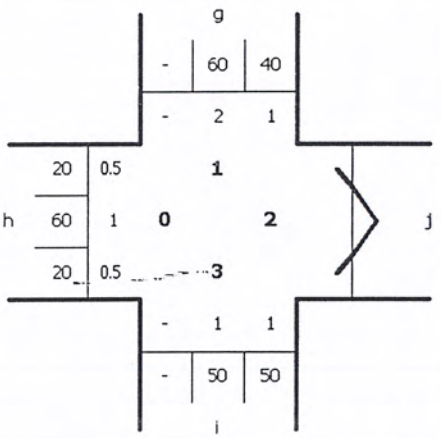
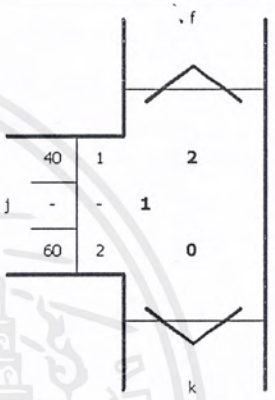
f	500	30	2	30	-	-	-
g	500	60	3	40	60	3	40
h	300	40	2	20	-	-	-
i	300	30	2	10	40	2	30
j	1000	60	3	20	-	-	-
k	300	30	2	30	-	-	-
l	400	25	2	30	-	-	-
m	300	50	3	15	-	-	-

ตารางที่ 5-3 แสดงรายละเอียดของถนนในแผนที่ที่ 2

ส่วนรายละเอียดของทางแยก แสดงได้ดังตาราง

ชื่อของทางแยก	จังหวะของสัญญาณไฟจราจร	เวลา (วินาที)	ความเร็ว (Km/h)	รายละเอียด
A	"23:20:21' "30:31"	30 30	15	
B	"13:31:12" "32:10" "20:21"	40 30 30	20	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C	"01:02" "32:31" "12:13"	30 30 30	20	
D	"10:12"	30	-1	

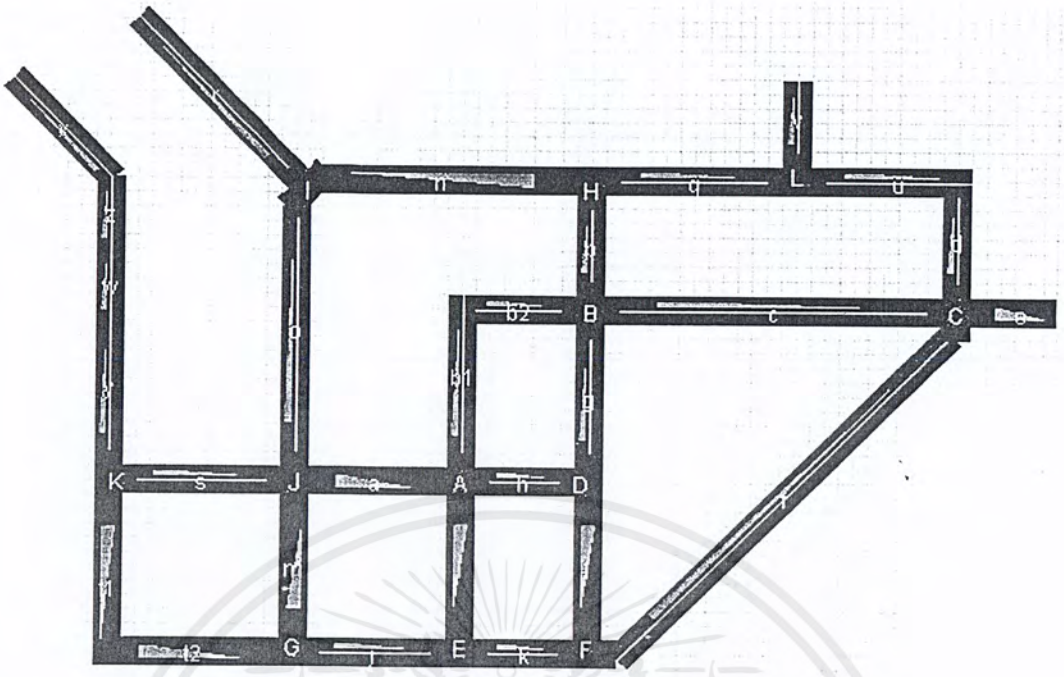
ตารางที่ 5-4 แสดงรายละเอียดของทางแยกในแผนที่ที่ 2

จากแผนที่ที่ 2 นั้นสามารถใช้งานแผนที่ได้โดยเรียกใช้งานไฟล์ Map_B.trf ภายในโฟลเดอร์ Save/TestMap เช่นเดียวกับไฟล์ Map_A.trf

แผนที่ที่ 3 จะเป็นแผนที่ที่มีขนาดใหญ่ และมีการทำงานของทางแยกที่สลับซับซ้อนกว่าแผนที่ที่ 2 โดยแผนที่ที่ 3 จะมีรายละเอียดของแผนที่เป็นดังรูปที่ 5-5

แผนที่ที่ 3 นั้นจะเน้นการทำงานในส่วนของถนน f ซึ่งเป็นถนนที่กว้างและมีความเร็วที่ใช้ในการวิ่งสูง หรืออาจจะเรียกว่าเป็น By pass way ก็ได้ โดยจำลองการทำงานในส่วนของตัวเมือง (ทางด้านซ้ายของแผนที่) และมีทางด่วนหรือทางเชื่อมจากภายนอกเข้ามา ซึ่งแผนที่ชนิดนี้สามารถเห็นได้ตามชานเมือง และมีการติดขัดของการจราจรมากในเขตเมือง และจะมีส่วนของถนน w ซึ่งเป็นถนนที่มีเขตชุมชนอยู่ โดยสมมติว่ามีการจอดรถที่ริมทางโดยตลอด (มองว่าความกว้างของถนนน้อยลงไป 1 เลนทั้งทางด้านไปและกลับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-5 แสดงลักษณะของแผนที่ที่ 3

รายละเอียดของถนนนั้นสามารถแสดงได้ตาราง

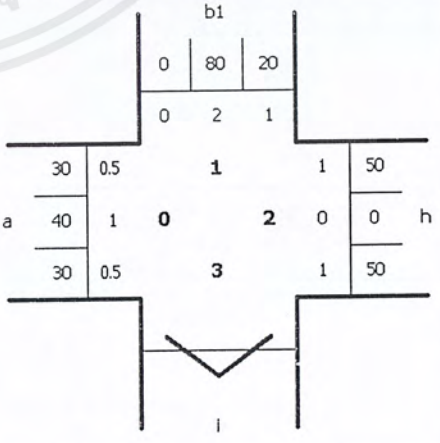
ชื่อของถนน	ความยาว (เมตร)	ด้าน Head ไปยังด้าน Foot			ด้าน Foot ไปยังด้าน Head		
		ความเร็ว (Km/h)	ความกว้าง (เลน)	ปริมาณรถเริ่มต้น	ความเร็ว (Km/h)	ความกว้าง (เลน)	ปริมาณรถเริ่มต้น
a	800	30	2	10	-	-	-
b1	1100	50	3	20	40	3	20
b2	600	60	3	10	40	2	15
c	2200	40	2	30	55	2	30
d	500	40	3	10	20	1	12
e	500	50	3	20	-	-	-
f	3250	70	3	5	70	3	5
g	800	30	2	10	35	2	15
h	500	30	3	10	40	2	20
i	800	40	2	10	-	-	-
j	800	40	1	30	40	3	10
k	500	30	1	10	50	3	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

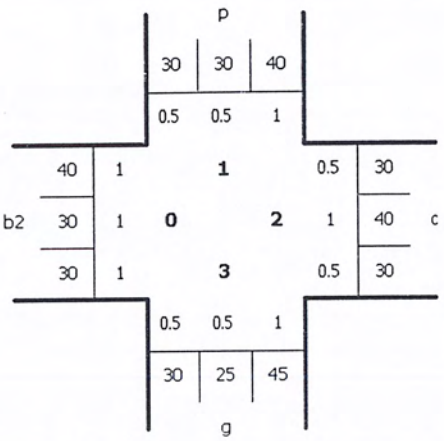
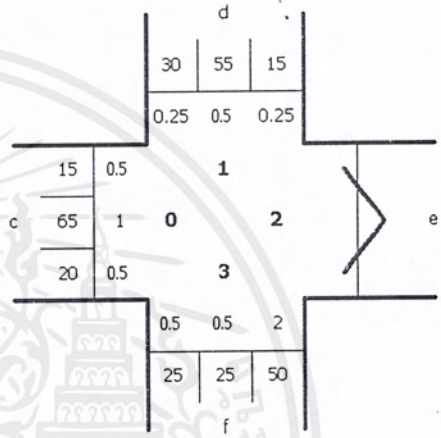
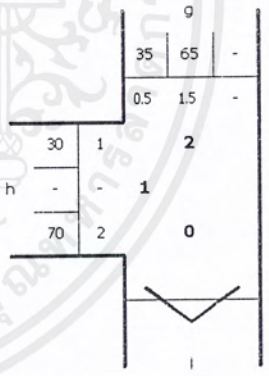
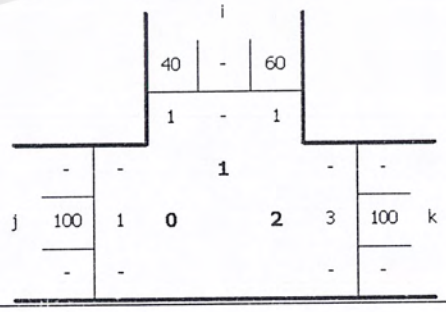
l	800	30	2	20	-	-	-
m	800	40	3	20	-	-	-
n	1700	70	2	30	-	-	-
o	1700	40	2	20	30	2	12
p	500	30	2	10	30	2	12
q	1000	30	3	20	45	2	10
r	1500	30	3	10	30	2	15
s	900	20	2	10	25	2	30
t1	1100	20	2	30	-	-	-
t2	1000	40	3	0	-	-	-
u	1000	20	3	30	20	2	10
v	500	30	2	10	30	3	40
w	500	15	2	40	15	2	40
x	1000	40	3	0	40	3	30
y	900	30	3	20	30	3	20
z	500	30	3	20	30	3	20

ตารางที่ 5-5 แสดงรายละเอียดของถนนในแผนที่ที่ 3

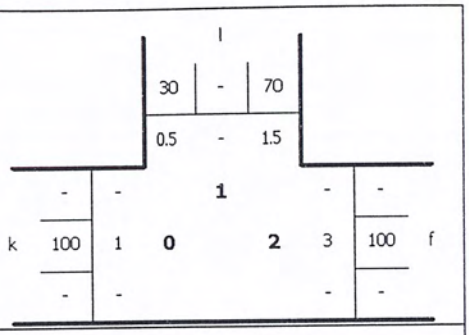
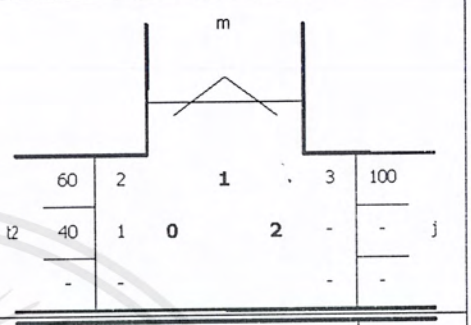
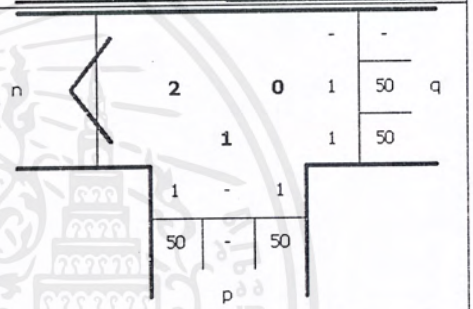
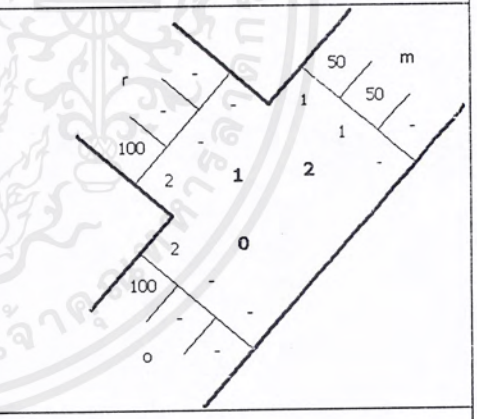
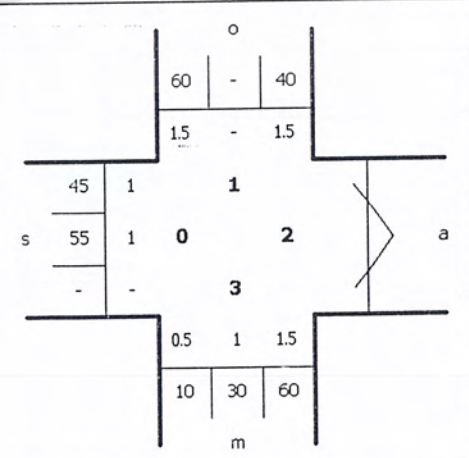
รายละเอียดของทางแยกแสดงได้ดังตาราง

ชื่อของทางแยก	จังหวะของสัญญาณไฟจราจร	เวลา (วินาที)	ความเร็ว (Km/h)	รายละเอียด
A	"01:13" "03:12:21" "13:23:02"	40 40 30	30	

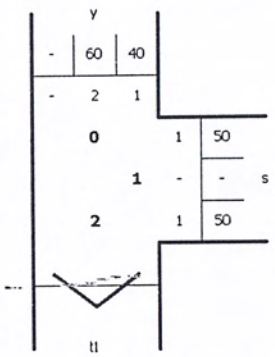
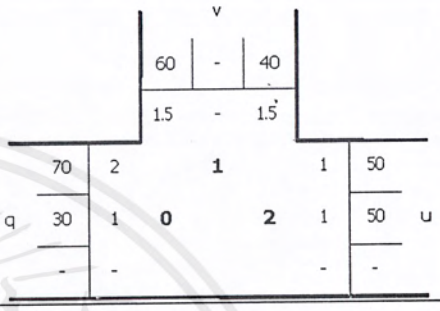
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B	"13:31:01:12" "02:20:12" "03:21:30:23" "10:32:01"	30 30 30 30	20	
C	"01:02:03" "10:12:13" "30:31:32"	30 20 40	40	
D	"12:10" "20:21'	40 40	30	
E	"12:20" "02:12"	30 30	30	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F	“02:20” “12:10”	50 20	20	
G	“01:02:21”	-1	15	
H	“01:10” “02:12”	30 30	25	
I	“01:10” “21:20”	40 30	25	
J	“01:02:30” “10:12:30:32” “31:32”	30 40 30	30	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K	"02:01" "10:12"	40 30	15	
L	"02:20:21" "01:10:12"	30 30	20	

ตารางที่ 5-6 แสดงรายละเอียดของแผนที่ที่ 3

แผนที่ที่ 3 นั้นจะอยู่ในไฟล์ที่ชื่อ Map_C.trf ซึ่งอยู่ในโฟลเดอร์เดียวกับ Map_A.trf และ Map_B.trf

สำหรับเครื่องที่นำมาใช้ในการทดสอบดังกล่าวนั้นมีลักษณะของเครื่องเป็นดังนี้

- หน่วยประมวลผลกลาง (CPU) คือ Pentium II 400 MHz
- หน่วยความจำ (RAM) คือ 128 Mbytes FSB 100 MHz
- ระบบปฏิบัติการที่ใช้ คือ Windows Me
- ฮาร์ดดิสก์ที่ใช้ คือ Quantum Fireball 8.4 Gbytes
- โปรแกรมที่ใช้ในการทำงาน คือ Java Runtime Environment เวอร์ชัน 1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.1 จำลองการทำงานของสัญญาณไฟจราจรแบบปรกติ (ตั้งค่าตามเวลา)

การจำลองระบบสัญญาณไฟจราจรแบบปรกตินั้นเราจะได้ผลการทดลองเรียงตามเวลาการทำงานของระบบออกมาเป็นดังนี้

แผนที่ที่ 1 จะได้ผลการทำงานเป็นดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	0	0	0	0	0
500	Car Rate	0.1397	0.3111	2.3908	2.5705	2.6073
	รถเข้า	292.8923	585.7846	2101.7691	2234.5809	2267.9835
	รถออก	223.0092	430.1894	906.3284	949.3284	964.3284
	เปอร์เซ็นต์	3.2445	7.2240	55.5025	59.6724	60.5268
1000	Car Rate	0.0713	0.1567	1.1949	1.2851	1.3041
	รถเข้า	585.2	1170.4000	3236.3842	3375.5806	3409.6497
	รถออก	513.8553	1013.6509	2041.4712	2090.4712	2105.4712
	เปอร์เซ็นต์	3.3124	7.2776	55.4781	59.6658	60.5511
2000	Car Rate	0.0346	0.0775	0.5976	0.6428	0.6520
	รถเข้า	1169.8154	2339.6308	5505.9649	5660.3937	5693.7958
	รถออก	1100.5475	2184.5741	4310.7569	4374.7569	4389.7569
	เปอร์เซ็นต์	3.2160	7.1990	55.4918	59.6903	60.5447
4000	Car Rate	0.0172	0.0386	0.2984	0.3213	0.3260
	รถเข้า	2339.0461	4678.0923	10044.0750	10225.5788	10259.6468
	รถออก	2269.9320	4523.4203	8850.3282	8940.3281	8955.3282
	เปอร์เซ็นต์	3.2088	7.1811	55.4240	59.6724	60.5577
6000	Car Rate	0.0118	0.0258	0.1993	0.2142	0.2171
	รถเข้า	3508.2769	7016.5539	14582.8629	14794.2688	14827.0029
	รถออก	3437.3165	6861.2665	13386.8995	13508.8994	13523.8995
	เปอร์เซ็นต์	3.2945	7.2097	55.5270	59.6780	60.5013
8000	Car Rate	0.0088	0.0194	0.1493	0.1606	0.1629

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รถเข้า	4677.5077	9355.0155	19120.9960	19359.5763	19392.9764
	รถออก	4606.7010	9199.1128	17926.4709	18074.4707	18089.4708
	เปอร์เซ็นต์	3.2874	7.2383	55.4602	59.6658	60.5201
	Car Rate	0.0069	0.0158	0.1197	0.1287	0.1306
10000	รถเข้า	5846.1539	11692.3078	23658.8916	23924.6862	23958.2964
	รถออก	5776.6578	11534.2490	22461.0251	22637.0250	22652.0250
	เปอร์เซ็นต์	3.2266	7.3384	55.6154	59.7845	60.6485

ตารางที่ 5-7 แสดงผลการทำงานของสัญญาณไฟจราจรแบบปรกติในแผนที่ที่ 1

แผนที่ที่ 2 ได้ผลการทดลองดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	15.7681	15.7681	15.7681	15.7681	15.7681
500	Car Rate	-0.7372	-0.6872	-0.0329	0.3733	0.4539
	รถเข้า	92.4923	184.9846	873.4358	1131.6991	1183.7303
	รถออก	461.1112	528.6169	889.8896	945.0450	956.7783
	เปอร์เซ็นต์	1.6543	3.2591	15.3105	23.2703	24.8871
1000	Car Rate	-0.4250	-0.3893	0.0105	0.2036	0.2339
	รถเข้า	184.8000	369.6000	1665.7435	1981.4684	2023.3263
	รถออก	609.8593	758.9316	1655.1987	1777.7729	1789.3730
	เปอร์เซ็นต์	1.6476	2.9191	16.4864	24.6073	25.8214
2000	Car Rate	-0.2548	-0.2307	0.0034	0.1122	0.1298
	รถเข้า	369.4153	738.8307	3250.3588	3697.6993	3745.9041
	รถออก	879.1176	1200.2564	3243.4803	3473.1899	3486.2258
	เปอร์เซ็นต์	2.1365	2.9493	16.9013	25.7993	27.2104
4000	Car Rate	-0.1726	-0.1559	0.0104	0.0994	0.1099
	รถเข้า	738.6461	1477.2923	6419.5895	7142.4688	7191.6429
	รถออก	1429.2125	2101.1037	6377.6584	9744.5188	6751.8522

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เปอร์เซ็นต์	1.6511	2.6521	19.1325	33.4334	35.1122
6000	Car Rate	-0.1453	-0.1293	0.0211	0.1039	0.1095
	รถเข้า	1107.8769	2215.7538	9588.8202	10581.9306	10622.8315
	รถออก	1979.7354	2992.0949	9461.6755	9958.4639	9965.6085
	เปอร์เซ็นต์	1.7251	2.8145	23.7940	43.4152	44.6909
8000	Car Rate	-0.1317	-0.1147	0.0325	0.0787	0.0821
	รถเข้า	1477.1077	2954.2154	12758.0509	13568.3922	13599.9647
	รถออก	2531.2772	3872.0154	12497.7992	12938.2771	12942.8913
	เปอร์เซ็นต์	1.5522	3.1564	29.5922	43.4466	44.5283
10000	Car Rate	-0.1211	-0.1068	0.0384	0.0606	0.0633
	รถเข้า	1846.1538	3692.3077	15925.8969	16539.6231	16568.5267
	รถออก	3057.3222	4761.1344	15541.8404	15932.8559	15934.8560
	เปอร์เซ็นต์	1.6069	2.6828	35.5988	42.9122	43.9114

ตารางที่ 5-8 แสดงผลการทำงานของสัญญาณไฟจราจรแบบปรกติในแผนที่ที่ 2

แผนที่ที่ 3 จะได้ผลการทดลองดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	4.7383	4.7383	4.7383	4.7383	4.7383
500	Car Rate	-0.1220	0.4540	4.3821	4.5855	4.5855
	รถเข้า	289.0384	578.0769	2545.1538	2646.8461	2646.8461
	รถออก	350.0767	351.0767	354.0767	354.0767	354.0767
	เปอร์เซ็นต์	4.3339	5.9092	16.7037	17.2610	17.2610
1000	Car Rate	-0.0061	0.5413	3.6536	3.7048	3.7048
	รถเข้า	577.500	1155.0000	4334.4870	4385.6703	4385.6703
	รถออก	583.6678	613.6877	680.8081	680.8081	680.8081
	เปอร์เซ็นต์	4.5055	7.5061	24.5695	24.8500	24.8500
2000	Car Rate	0.1064	0.6391	2.2446	2.2448	2.2448

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รถเข้า	1154.4230	2308.8461	5682.3396	5682.7715	5682.7715
	รถออก	941.5355	1030.5011	1193.0093	1193.0093	1193.0093
	เปอร์เซ็นต์	5.7914	11.3495	28.9178	28.9202	28.9202
4000	Car Rate	0.2007	0.6614	1.3226	1.3240	1.3240
	รถเข้า	2308.2692	4294.9468	7159.9380	7165.7917	7165.7917
	รถออก	1505.1466	1649.2733	1869.4735	1869.4735	1869.4735
	เปอร์เซ็นต์	9.0791	18.6410	33.0459	33.0780	33.0780
6000	Car Rate	0.2350	0.5886	0.9247	0.9247	0.9247
	รถเข้า	3462.1154	5763.1523	7961.1253	7961.1227	7961.1227
	รถออก	2051.6632	2231.4758	2412.8760	2412.8769	2412.8769
	เปอร์เซ็นต์	12.6374	23.3123	34.8476	34.8476	34.8476
8000	Car Rate	0.2455	0.5295	0.6698	0.6698	0.6698
	รถเข้า	4533.5237	6959.6501	7692.3854	7962.3749	7962.3749
	รถออก	2569.4040	2722.9144	2603.3257	2603.2149	2603.2149
	เปอร์เซ็นต์	15.7522	27.2096	35.8495	35.8645	35.8645
10000	Car Rate	0.2257	0.4505	0.5190	0.5188	0.5188
	รถเข้า	5332.0249	7439.5676	7962.7423	7692.7183	7692.7183
	รถออก	3074.2112	2933.7942	2772.0601	2773.7694	2773.7694
	เปอร์เซ็นต์	17.5390	29.4004	37.4189	37.4427	37.4427

ตารางที่ 5-9 แสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบปรกติในแผนที่ที่ 3

จากตารางทั้งสามนั้น จะเห็นได้ว่าถ้าปริมาณรถที่เข้าสู่ถนนนั้นมีค่ามากๆ ปริมาณรถที่เข้าสู่ระบบจะมากเฉพาะในช่วงแรก และจะเริ่มคงที่เมื่อมีการทำงานต่อมาเรื่อยๆ นั่นเป็นเพราะว่าระบบนั้นมาถึงจุดอิ่มตัว (Critical Point) ของระบบแล้ว ปริมาณรถที่ออกจากระบบกับปริมาณของรถที่เข้าสู่ระบบจะเริ่มคงที่ ดังนั้นระบบสัญญาณไฟจราจรที่ดีนั้นจะสามารถรองรับการทำงานของรถได้มาก (มีปริมาณรถที่เข้าและออกจากระบบมาก)

ส่วนการทำงานของระบบในส่วนของอัลกอริทึมเลือกเวลาที่ดีที่สุด (Best Time) จะได้ผลดังหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ผลการทดลองระบบสัญญาณไฟจราจรแบบเลือกจังหวะสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุด (Best Select)

ผลการทำงานแบบเลือกจังหวะสัญญาณไฟจราจรที่ดีที่สุดนั้นแสดงได้ดังตารางด้านล่าง โดยจะมีการเลือกทางแยกทั้งหมดที่อยู่ในระบบ

แผนที่ที่ 1 จะได้ผลการทำงานแสดงได้ดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	0	0	0	0	0
500	Car Rate	0.1362	0.2705	2.4894	2.5996	2.6000
	รถเข้า	292.8923	585.7846	2304.3738	2388.2488	2362.6806
	รถออก	224.7615	450.5015	1059.6319	1088.4392	1062.6795
	เปอร์เซ็นต์	3.1632	6.2810	57.7915	60.3483	60.3571
1000	Car Rate	0.0670	0.1338	1.2980	1.3096	1.3227
	รถเข้า	585.2000	1170.4000	3593.3844	3641.1533	3639.1025
	รถออก	518.1645	1036.5227	2295.2892	2331.5240	2316.3940
	เปอร์เซ็นต์	3.1123	6.2157	60.2687	60.8042	61.4114
2000	Car Rate	0.0369	0.0663	0.6409	0.6488	0.6601
	รถเข้า	1169.8154	2339.6308	6090.7810	6160.5373	6163.0444
	รถออก	1095.9860	2207.0153	4808.8765	4862.7964	4842.6887
	เปอร์เซ็นต์	3.4277	6.1571	59.5168	60.2523	61.3022
4000	Car Rate	0.0169	0.0336	0.3136	0.3251	0.3246
	รถเข้า	2339.0461	4678.0923	11037.0754	11240.3695	11215.3462
	รถออก	2271.3383	4543.4546	9782.3178	9939.8204	9916.9332
	เปอร์เซ็นต์	3.1435	6.2510	58.2566	60.3827	60.2835
6000	Car Rate	0.0114	0.0228	0.2152	0.2195	0.2194
	รถเข้า	3508.2769	7016.5539	16074.8865	16300.5348	16290.8507
	รถออก	3439.7692	6879.6444	14783.2243	14983.2922	14973.9332
	เปอร์เซ็นต์	3.1807	6.3565	59.9701	61.1579	61.1428

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8000	Car Rate	0.0084	0.0170	0.1604	0.1631	0.1654
	รถเข้า	4677.5077	9355.0155	21085.8969	21338.3027	21339.4051
	รถออก	4609.6535	9218.5967	19802.2646	20033.2060	20015.6137
	เปอร์เซ็นต์	3.1503	6.3337	59.5974	60.5940	61.4620
10000	Car Rate	0.0069	0.0133	0.1264	0.1292	0.1306
	รถเข้า	5846.1539	11692.3078	26041.8848	26398.7422	26384.2946
	รถออก	5777.0688	11559.0859	24776.9785	25106.4344	25077.8752
	เปอร์เซ็นต์	3.2075	6.1853	58.7280	60.0003	60.6555

ตารางที่ 5-10 แสดงผลการทำงานของสัญญาณไฟจราจร
แบบเลือกจังหวะสัญญาณไฟจราจรที่ปล่อยรถได้มากที่สุดในแผนที่ที่ 1

แผนที่ที่ 2 จะได้ผลการทำงานเป็นดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	15.7681	15.7681	15.7681	15.7681	15.7681
500	Car Rate	-0.7051	-0.6620	-0.2388	0.4929	0.7434
	รถเข้า	92.4923	184.9846	924.9230	1519.3029	1468.4996
	รถออก	445.0798	516.0025	1044.3236	1272.8117	1096.7542
	เปอร์เซ็นต์	1.2335	2.2541	10.9106	25.6710	30.7014
1000	Car Rate	-0.3506	-0.3343	-0.1077	0.4071	0.7523
	รถเข้า	184.8000	369.6000	1848.0000	2700.6992	2538.8444
	รถออก	535.4308	703.9417	1955.7516	2293.5574	1786.5005
	เปอร์เซ็นต์	1.1185	2.2253	11.1329	32.1170	45.9880
2000	Car Rate	-0.1756	-0.1711	-0.0440	0.3628	0.4480
	รถเข้า	369.4153	738.8307	3694.1538	4133.8317	3380.0518
	รถออก	720.8008	1081.1232	3782.2513	3408.2135	2484.0357
	เปอร์เซ็นต์	1.3604	2.3394	11.3470	47.4472	51.7997
4000	Car Rate	-0.1015	-0.0837	-0.0162	0.2693	0.2672

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รถเข้า	738.6461	1477.2923	7386.4615	5831.4108	5138.6461
	รถออก	1144.7267	1812.2357	7451.6098	4754.1945	4069.8180
	เปอร์เซ็นต์	1.3081	2.3368	11.2807	63.0310	58.8119
6000	Car Rate	-0.0745	-0.0582	-0.0069	0.1825	0.2032
	รถเข้า	1107.8769	2215.7538	11078.7692	7664.9456	6916.4454
	รถออก	1555.1879	2565.4029	11120.2498	6569.5748	5696.9212
	เปอร์เซ็นต์	1.3006	2.3702	10.9476	63.6843	65.4083
8000	Car Rate	-0.0593	-0.0453	-0.0017	0.1275	0.1429
	รถเข้า	1477.1077	2954.2154	14771.0769	9449.5829	8777.3870
	รถออก	1952.1136	3317.0915	14784.6994	8429.4057	7633.9255
	เปอร์เซ็นต์	1.3983	2.2455	10.9442	60.6036	62.5263
10000	Car Rate	-0.0476	-0.0361	0.0011	0.1128	0.1008
	รถเข้า	1846.3385	3692.6770	18461.5385	11287.6361	10587.3048
	รถออก	2322.5912	4054.4055	18449.8311	10158.7457	9578.7311
	เปอร์เซ็นต์	1.3928	2.3052	10.4620	64.9847	57.5592

ตารางที่ 5-11 แสดงผลการทำงานของสัญญาณไฟจราจร
แบบเลือกจังหวะสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุดในแผนที่ที่ 2

แผนที่ที่ 3 จะได้ผลการทำงานดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	4.7383	4.7383	4.7383	4.7383	4.7383
500	Car Rate	-0.1544	0.4182	4.3396	4.8148	4.8149
	รถเข้า	289.0384	578.0769	2550.1623	2788.7179	2788.7179
	รถออก	366.2585	368.9508	380.3504	381.2820	381.2649
	เปอร์เซ็นต์	4.2678	5.8453	16.6102	17.9280	17.9281
1000	Car Rate	-0.0178	0.5265	3.6349	3.7566	3.7587
	รถเข้า	577.5000	1155.0000	4358.8727	4506.0454	4506.0454

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รถออก	595.3499	628.4422	723.9032	749.4161	747.2798
	เปอร์เซ็นต์	4.3289	7.3571	24.5772	25.2141	25.1959
2000	Car Rate	0.1249	0.6619	2.1686	2.1141	2.1719
	รถเข้า	1154.4230	2308.8461	5563.9727	5457.4181	5590.3255
	รถออก	904.4732	984.8662	1226.7022	1229.0492	1246.3837
	เปอร์เซ็นต์	5.3973	11.3882	28.1172	27.5264	28.0119
4000	Car Rate	0.2376	1.0987	1.2171	1.1838	1.2191
	รถเข้า	2308.2692	4395.1099	6584.7819	6533.1307	6657.1614
	รถออก	1357.6585	1515.5681	1716.1459	1797.5333	1780.4634
	เปอร์เซ็นต์	8.5916	19.3226	30.6378	29.9479	30.3394
6000	Car Rate	0.2864	0.6338	0.9007	0.8365	0.8506
	รถเข้า	3462.1154	5810.4958	7499.1548	7109.6978	7205.7009
	รถออก	1743.4264	2007.3022	2094.4177	2090.5013	2102.0295
	เปอร์เซ็นต์	12.2847	23.8683	34.1307	32.2981	32.5676
8000	Car Rate	0.3104	0.5275	0.6724	0.6389	0.6279
	รถเข้า	4615.9615	6683.2063	7574.2995	7379.3794	7299.0337
	รถออก	2132.4962	2462.6554	2194.3836	2267.4069	2275.7630
	เปอร์เซ็นต์	15.9442	26.3371	35.0741	33.9261	33.9728
10000	Car Rate	0.3256	0.4462	0.5346	0.5061	0.4971
	รถเข้า	5769.2307	7215.8999	7575.0593	7459.5156	7400.9222
	รถออก	2513.1323	2753.1849	2228.1080	2398.1418	2429.8484
	เปอร์เซ็นต์	20.1080	28.9165	36.0426	35.1011	35.2586

ตารางที่ 5-12 แสดงผลการทำงานของสัญญาณไฟจราจร
แบบเลือกจังหวะสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุดในแผนที่ที่ 3

จากผลการทำงานของระบบสัญญาณไฟจราจรด้านบนนั้นจะเห็นได้ว่า การทำงานของสัญญาณไฟจราจรในแบบนี้นั้น จะทำงานได้ดีเมื่อมีปริมาณรถเข้าสู่ระบบในปริมาณมาก ซึ่งจะปล่อยปริมาณรถได้มากด้วย แต่ว่าก็จะมีค่ามากในระดับหนึ่งเท่านั้น เมื่อมีค่าที่มากเกินไป หรือ น้อยเกินไปก็จะทำให้การทำงานของสัญญาณไฟจราจรในแบบนี้นั้นมีการทำงานที่ไม่แต่ต่างกับการทำงานในโหมดของสัญญาณไฟจราจรแบบปรกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของระบบสัญญาณไฟจราจรในแบบนี้คือ ปริมาณของ Car Rate (อัตราการวิ่งของรถในระบบ) จะดีเพราะว่าจะปล่อยเฉพาะสัญญาณไฟจราจรที่ทำให้รถวิ่งได้มากที่สุดเท่านั้น ซึ่งก็จะทำให้การทำงานของระบบโดยรวมดีขึ้น

ข้อเสียของระบบสัญญาณไฟจราจรในแบบนี้คือ ในกรณีที่มีรถในถนนเส้นนั้นๆ น้อย ก็จะทำให้รถในถนนนั้นๆ รอนานมากๆ หรือบางที่อาจจะไม่มีโอกาสในการวิ่งเลยก็ได้

การปรับปรุงแก้ไขนี้นสามารถทำได้โดยการสร้างส่วนของการตรวจวัดเวลาของการรอของถนนในทางแยกว่ามีเวลาเป็นเท่าใด ถ้าเกิดรอนานมากๆ ก็ปล่อยสัญญาณไฟจราจรในถนนนั้นก่อนที่จะเลือกสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุด หรือกำหนดสัญญาณไฟจราจรใหม่ โดยให้จังหวะของสัญญาณไฟจราจรนั้นเล็กที่สุด เพื่อให้การคำนวณเวลาของการปล่อยสัญญาณไฟจราจรนั้นเป็นไปอย่างถูกต้อง (เพราะว่าในการเลือกจังหวะสัญญาณไฟจราจรนั้นจะพิจารณาสัญญาณไฟจราจรทุกอันมีความสำคัญเท่ากันหมด ในกรณีที่รถที่อยู่ในถนนมี 100 กับ 5 คัน เวลาของการปล่อยสัญญาณไฟจราจรนั้นก็จะใช้ 5 คันมาเป็นตัวคิด เพื่อให้สัญญาณไฟจราจรที่ปล่อยนั้นมีรถวิ่งได้มากที่สุดนั่นเอง)

5.2.3 ผลการทำงานของระบบสัญญาณไฟจราจรแบบเลือกคำนวณเวลาของสัญญาณไฟจราจรที่ดีที่สุด (Best Time)

ผลการทำงานของระบบสัญญาณไฟจราจรแบบคำนวณเวลาของสัญญาณไฟจราจรที่ดีที่สุดนั้นสามารถแสดงผลการทำงานได้ดังนี้

แผนที่ที่ 1 จะได้ผลการทำงานแสดงได้ดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	0	0	0	0	0
500	Car Rate	0.1345	0.2691	2.4253	2.6261	2.6517
	รถเข้า	292.8923	585.7846	2151.0682	2217.1194	2257.0518
	รถออก	225.5923	451.2102	2151.0682	904.0258	931.1686
	เปอร์เซ็นต์	3.1246	6.248	56.3022	60.9650	61.5588
1000	Car Rate	0.069	0.1373	1.2503	1.3130	1.3265
	รถเข้า	585.2	1170.4	5657.1124	3358.1191	3398.7179
	รถออก	516.2	1033.0071	4363.1584	2045.0286	2072.1714

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เปอร์เซ็นต์	3.2057	6.3789	60.0764	60.9649	61.5896
2000	Car Rate	0.034	0.0666	0.6469	0.6564	0.6628
	รถเข้า	1172.7384	2339.3753	10201.075	5641.9320	5681.8639
	รถออก	1104.5368	2206.3753	8940.3001	4329.0327	4356.1756
	เปอร์เซ็นต์	3.1665	6.1868	58.5360	60.9560	61.5498
4000	Car Rate	0.0173	0.0338	0.3151	0.3282	0.3316
	รถเข้า	2339.0461	4678.0923	14766.9388	10208.1172	10248.7151
	รถออก	2269.8384	4542.6768	13558.5683	8895.0432	8922.1860
	เปอร์เซ็นต์	3.2132	6.2871	56.103	60.9642	61.5889
6000	Car Rate	0.0114	0.0227	0.2013	0.2187	0.2209
	รถเข้า	3508.2769	7016.5539	19408.9464	14775.8071	14816.0710
	รถออก	3439.6906	6879.7931	18125.643	13463.0529	13490.1957
	เปอร์เซ็นต์	3.1843	6.3496	59.5821	60.9494	61.5586
8000	Car Rate	0.0086	0.017	0.1604	0.1641	0.1657
	รถเข้า	4677.5077	9355.0155	19408.9464	19342.1148	19382.0446
	รถออก	4608.16	9218.3201	18125.643	18029.0634	18056.2062
	เปอร์เซ็นต์	3.2197	6.3465	59.5821	60.9633	61.5570
10000	Car Rate	0.007	0.0137	0.1271	0.1314	0.1327
	รถเข้า	5846.1539	11692.3078	23965.6872	23907.2247	23947.3647
	รถออก	5776.0919	11554.9875	22694.3158	22592.3381	22619.4809
	เปอร์เซ็นต์	3.2528	6.3755	59.0282	61.0485	61.6520

ตารางที่ 5-13 แสดงผลการทำงานของระบบสัญญาณไฟจราจร
แบบเลือกคำนวณเวลาของสัญญาณไฟจราจรที่ดีที่สุดในแผนที่ที่ 1

แผนที่ที่ 2 แสดงผลการทำงานได้ดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	15.7681	15.7681	15.7681	15.7681	15.7681

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

500	Car Rate	-0.7002	-0.6522	-0.2280	0.4784	0.5573
	รถเข้า	92.4923	184.9846	924.923	1495.6051	1537.1965
	รถออก	445.0798	511.1043	1038.9444	1256.3731	1258.5402
	เปอร์เซ็นต์	1.1851	2.2211	10.871	25.3798	26.9164
1000	Car Rate	-0.3354	-0.3130	-0.1031	0.2219	0.2824
	รถเข้า	184.8	369.6	1848	2702.0838	2795.2982
	รถออก	520.2995	682.6515	1951.1801	2480.1331	2512.7987
	เปอร์เซ็นต์	1.1187	2.2133	10.9912	24.6864	27.0706
2000	Car Rate	-0.1502	-0.1444	-0.0419	0.1667	0.1385
	รถเข้า	369.4153	738.8307	3694.1538	4797.3917	5124.4430
	รถออก	369.4153	1027.7726	3777.9715	4463.9555	4847.2718
	เปอร์เซ็นต์	1.1643	2.2012	10.9951	29.0526	26.8620
4000	Car Rate	-0.0619	-0.0584	-0.0108	0.1750	0.1800
	รถเข้า	738.6461	1477.2923	7386.4615	8479.6772	9183.5479
	รถออก	986.6168	1711.2849	7429.7631	7779.4168	8463.4780
	เปอร์เซ็นต์	1.2336	2.2216	10.9317	43.1372	44.3360
6000	Car Rate	-0.0323	-0.0305	-8.3129E-4	0.1227	0.1256
	รถเข้า	1107.8769	2215.7538	11078.7692	11522.6797	12229.5241
	รถออก	1301.8261	2398.7619	11083.7570	10785.9415	11475.8230
	เปอร์เซ็นต์	1.053	2.1077	10.8713	43.1563	44.4304
8000	Car Rate	-0.0180	-0.0155	0.0049	0.0963	0.0989
	รถเข้า	1477.1077	2954.2154	14771.0769	14564.6824	15273.0415
	รถออก	1621.2950	3078.2361	14731.1212	13794.0017	14481.6969
	เปอร์เซ็นต์	1.1459	2.2188	10.9180	43.1540	44.4677
10000	Car Rate	-0.0092	-0.0065	0.0078	0.0803	0.0826
	รถเข้า	1846.3385	3692.6770	18461.5385	17607.6852	18316.1888
	รถออก	1938.7077	3757.8429	18382.5627	16804.4534	17489.3656
	เปอร์เซ็นต์	1.1304	2.1375	11.0171	43.1359	44.4580

ตารางที่ 5-14 แสดงผลการทำงานของระบบสัญญาณไฟจราจร
แบบคำนวณเวลาของสัญญาณไฟจราจรที่ดีที่สุดในแผนที่ที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนที่ที่ 3 แสดงผลการทำงานได้ดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	4.7383	4.7383	4.7383	4.7383	4.7383
500	Car Rate	-0.1529	0.4246	4.2292	4.3906	4.3906
	รถเข้า	289.0384	578.0769	2501.9230	2585.6153	2585.6153
	รถออก	365.5096	365.7476	387.3213	390.2796	390.2796
	เปอร์เซ็นต์	3.6587	5.2040	15.7149	16.2092	16.2092
1000	Car Rate	-0.0015	0.5387	3.4075	3.4115	3.4115
	รถเข้า	577.5000	1155.0000	4211.7178	4220.6408	4220.6408
	รถออก	579.0990	616.2683	804.1541	809.1046	809.1046
	เปอร์เซ็นต์	2.9171	5.9150	22.1896	22.1907	22.1907
2000	Car Rate	0.1594	0.5259	2.4686	2.4695	2.4695
	รถเข้า	1154.4230	2308.8461	6884.4932	6889.4931	6889.4931
	รถออก	835.5510	1256.9099	1947.2556	1950.4803	1950.4803
	เปอร์เซ็นต์	2.6313	6.4370	29.4530	29.4044	29.4044
4000	Car Rate	0.2599	0.5210	1.5188	1.4854	1.4854
	รถเข้า	2308.2692	4616.5384	9190.9233	9176.3664	9176.3663
	รถออก	1268.2981	2532.1426	3115.5315	3234.5098	3234.5098
	เปอร์เซ็นต์	2.4537	7.5202	34.4952	35.1669	35.1669
6000	Car Rate	0.3032	0.5284	1.0074	0.9634	0.9634
	รถเข้า	3462.1154	6924.2308	9190.9616	9176.3914	9176.3913
	รถออก	1642.4583	3753.6516	3145.9720	3395.7094	3395.7094
	เปอร์เซ็นต์	2.4971	8.8392	34.5827	36.7029	36.7029
8000	Car Rate	0.3231	0.5307	0.7531	0.7087	0.7087
	รถเข้า	4615.9615	9231.9231	9190.9992	9176.4163	9176.4163
	รถออก	2030.8809	4985.6984	3165.9302	3506.6739	3506.6739

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เปอร์เซ็นต์	2.4915	10.1849	34.7192	37.9096	37.9096
10000	Car Rate	0.3331	0.5341	0.6001	0.5556	0.5556
	รถเข้า	5769.2307	11538.4615	9191.0075	9176.4413	9176.4413
	รถออก	2437.9947	6196.4875	3189.9083	3620.0520	3620.0520
	เปอร์เซ็นต์	2.4359	11.5434	34.8841	38.4361	38.4361

ตารางที่ 5-15 แสดงผลการทำงานของระบบสัญญาณไฟจราจร
แบบคำนวณเวลาของสัญญาณไฟจราจรที่ดีที่สุดในแผนที่ที่ 3

จากตารางการทำงานของระบบสัญญาณไฟจราจรด้านบนนั้นจะเห็นได้ว่าการทำงานของสัญญาณไฟจราจรในแบบนี้นั้น เป็นการทำงานที่มีผลดีกว่าระบบสัญญาณไฟจราจรแบบตั้งค่าตามเวลา (Normal Mode) แต่ว่าจะมีผลกาทำงานด้อยกว่าแบบเลือกสัญญาณไฟจราจรที่ปล่อยปริมาณรถได้มากที่สุด (Best Select Mode) ในการทำงานในระบบนี้นั้นจะได้ข้อดีของระบบทั้งสองมาอย่างละครึ่ง คือ การปล่อยสัญญาณไฟจราจรเป็นไปอย่างแน่นอน ไม่มีการปล่อยสัญญาณไฟจราจรที่เสียเวลาเปล่า แต่อาจจะมีการสูญเสียเวลาของการทำงานไปบ้าง ในกรณีที่ปล่อยสัญญาณไฟจราจรที่ไม่มีรถวิ่งอยู่ในถนนเลย (โอกาสในการเป็นไปได้นั้นน้อยมาก)

ข้อดีของการทำงานของระบบสัญญาณไฟจราจรในแบบนี้คือ มีลำดับของการปล่อยสัญญาณไฟจราจรที่แน่นอน ซึ่งจะไม่เกิดกรณีที่รถรอนานๆ และจะมีการทำงานที่ง่ายและรวดเร็ว

ข้อเสียของการทำงาน คือ ในบางกรณีที่ปล่อยสัญญาณไฟจราจรไปแล้วไม่มีรถวิ่งอยู่ในถนนนั้นเลย (ถือว่าเป็น Worst Case ของการทำงานของระบบสัญญาณไฟจราจรในแบบนี้)

5.2.4 ผลการทำงานของระบบสัญญาณไฟจราจรแบบเลือกถนนที่ต้องการปล่อยรถได้มากที่สุด (Select Road Mode)

จากผลการทำงานของระบบสัญญาณไฟจราจรดังกล่าวจะต้องมีการเลือกถนนที่ต้องการด้วย ซึ่งจะได้ผลการทำงานเป็นดังตาราง ซึ่งในการทดลองสัญญาณไฟนั้น เราจะเลือกถนนที่อยู่ใกล้กับปลายเปิด (Open Node) และเลือกถนนที่อยู่บริเวณกลางแผนที่เพื่อเปรียบเทียบค่าที่ได้จากการทำงาน ดังนั้นเราจึงทดสอบการทำงานที่แผนที่ที่ 2 เท่านั้น โดยจะมีการเลือกถนนทั้งสองแบบดังที่ได้กล่าวมาแล้ว

แผนที่ที่ 2 ในกรณีที่เลือกถนนปลายเปิดแสดง (ในการทดสอบเลือกถนน d ในแผนที่ที่ 2 เป็นถนนสองทาง โดยมีปลายด้านหนึ่งเป็นปลายเปิด) ผลการทำงานได้ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0
	เปอร์เซ็นต์	15.7681	15.7681	15.7681	15.7681	15.7681
500	Car Rate	-0.6886	-0.5946	-0.1352	0.3051	0.3024
	รถเข้า	92.4923	184.9846	924.9230	1179.6564	1201.0294
	รถออก	436.8172	482.3201	992.5397	1027.0589	1049.8273
	เปอร์เซ็นต์	2.9637	4.3638	13.3526	22.3996	23.0448
1000	Car Rate	-0.4060	-0.3544	-0.0608	0.1501	0.1421
	รถเข้า	184.8000	369.6000	1848.0000	2096.1350	2107.3348
	รถออก	590.8575	724.0700	1908.8116	1945.9572	1965.2135
	เปอร์เซ็นต์	1.4421	2.9928	13.4998	22.3807	22.6360
2000	Car Rate	-0.2310	-0.1984	-0.0344	0.0689	0.0687
	รถเข้า	369.4153	738.8307	3694.1538	3945.6565	3963.2031
	รถออก	831.5278	1135.7943	3763.0678	3807.7447	3825.6047
	เปอร์เซ็นต์	2.4337	2.6299	12.9404	21.9649	22.3703
4000	Car Rate	-0.1497	-0.1186	-0.0199	0.0318	0.0353
	รถเข้า	738.6461	1477.2923	7386.4615	7657.1352	7675.6512
	รถออก	1337.8337	1951.9195	7466.2858	7529.8228	7534.2983
	เปอร์เซ็นต์	1.4431	2.5774	12.6971	22.0227	22.4427
6000	Car Rate	-0.1209	-0.0893	-0.0143	0.0204	0.0266
	รถเข้า	1107.8769	2215.7538	11078.7692	11363.3063	11393.6907
	รถออก	1833.3924	2751.7159	11164.8462	11240.4118	11233.9230
	เปอร์เซ็นต์	1.5557	2.6002	12.6115	22.2131	22.9890
8000	Car Rate	-0.1076	-0.0732	-0.0080	0.0155	0.0200
	รถเข้า	1477.1077	2954.2154	14771.0769	15066.3491	15077.4476
	รถออก	2338.4452	3540.5728	14835.7253	14941.9418	14916.6563
	เปอร์เซ็นต์	1.8129	2.8030	13.2298	22.7090	22.9502
10000	Car Rate	-0.0968	-0.0652	-0.0052	0.0110	0.0157

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รถเข้า	1846.1538	3692.3077	18461.5385	18760.3578	18771.7224
	รถออก	2814.4547	4344.6603	18513.6647	18649.4020	18614.7026
	เปอร์เซ็นต์	1.5322	2.5911	13.5312	22.4517	22.5919

ตารางที่ 5-16 แสดงผลการทำงานของระบบสัญญาณไฟจราจร
แบบเลือกเส้นทางที่ต้องการปล่อยรถได้มากที่สุดในแผนที่ที่ 2 (เลือกถนนปลายเปิด)

ส่วนค่าของปริมาณรถที่เข้าสู่ถนนที่เลือกในถนนปลายเปิดเป็นดังตาราง โดยเปรียบเทียบกับระบบสัญญาณไฟจราจรแบบปรกติได้ดังนี้

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	$Car\ Rate_{Normal}$	0, 0	0, 0	0, 0	0, 0	0, 0
	$Car\ Rate_{Select}$	0, 0	0, 0	0, 0	0, 0	0, 0
500	$Car\ Rate_{Normal}$	-0.0548, -0.0157	-0.0499, -0.0075	-0.0110, 0.1594	-0.0090, 0.1594	-0.0090, 0.1594
	$Car\ Rate_{Select}$	-0.0558, -0.0157	-0.0494, -0.0095	-0.0171, 0.0790	-0.0528, 0.1575	-0.0543, 0.1575
1000	$Car\ Rate_{Normal}$	-0.0276, -0.0077	-0.0242, -0.0045	-0.0065, 0.0797	-0.0045, 0.0797	-0.0045, 0.0797
	$Car\ Rate_{Select}$	-0.0279, -0.0067	-0.0247, -0.0045	-0.0117, 0.0330	-0.0244, 0.0787	-0.0255, 0.0787
2000	$Car\ Rate_{Normal}$	-0.0136, -0.0038	-0.0125, -0.0021	-0.0034, -0.0398	-0.0024, 0.0398	-0.0024, 0.0398
	$Car\ Rate_{Select}$	-0.0133, -0.0038	-0.0133, -0.0021	-0.0086, 0.0165	-0.0131, 0.0393	-0.0131, 0.0393
4000	$Car\ Rate_{Normal}$	-0.0066, -0.0018	-0.0063, -0.0011	-0.0017, 0.0199	-0.0012, 0.0199	-0.0012, 0.0199
	$Car\ Rate_{Select}$	-0.0069, -0.0023	-0.0066, -0.0014	-0.0031, 0.0086	-0.0065, 0.0196	-0.0068, 0.0196

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6000	$Car\ Rate_{Normal}$	-0.0043, -0.0013	-0.0041, -6.7685E-4	-0.0011, -0.0132	-8.0819E-4, 0.0132	-8.0819E-4, 0.0132
	$Car\ Rate_{Select}$	-0.0046, -0.0013	-0.0043, -3.4358E-4	-0.0026, 0.0049	-0.0046, 0.0131	-0.0043, 0.0131
8000	$Car\ Rate_{Normal}$	-0.0035, -9.6533E-4	-0.0030, -5.5569E-4	-8.5590E-4, 0.0099	-6.0590E-4, 0.0099	-6.0590E-4, 0.0099
	$Car\ Rate_{Select}$	-0.0034, -9.6538E-4	-0.0032, -8.0576E-4	-0.0029, 0.0032	-0.0031, 0.0098	-0.0031, 0.0098
10000	$Car\ Rate_{Normal}$	-0.0027, -8.4610E-4	-0.0025, -4.9223E-4	-6.7340E-4, 0.0079	-5.0589E-4, 0.0079	-5.0589E-4, 0.0079
	$Car\ Rate_{Select}$	--0.0026, -8.4615E-4	-0.0025, -2.9230E-4	-0.0022, 0.0030	-0.0024, 0.0078	-0.0025, 0.0078

ตารางที่ 5-17 แสดงผลของรายละเอียดถนนที่เลือก (ถนนปลายเปิด) ในแผนที่ที่ 2 เปรียบเทียบกับสัญญาณไฟจราจรแบบปรกติ

จากตารางจะเห็นได้ว่ามีค่าสองค่า โดยที่ค่าบนเป็นค่าของถนนที่เลือก (ถนน d) ที่ด้าน Head ไปยังด้าน Foot ส่วนค่าที่สองนั้นเป็นค่าของถนนที่เลือกจากด้าน Foot ไปยังด้าน Head (เป็นด้านที่มีการควบคุมสัญญาณไฟจราจรได้ ในการปล่อยสัญญาณไฟจราจรแบบ Select Road) และจากตารางจะเห็นได้ว่าค่าของ Car Rate นั้นมีค่าไม่แตกต่างกันมากในระหว่างการทำงานแบบสัญญาณไฟจราจรปรกติ กับสัญญาณไฟจราจรแบบเลือกถนนที่ต้องการ ทั้งนี้เนื่องมาจากว่าสัญญาณไฟจราจรที่ทางแยก B มีจังหวะของสัญญาณไฟจราจรแบบ 3 จังหวะและมีจังหวะของสัญญาณไฟจราจรที่ทำให้ถนน d เคลื่อนที่นั้นอยู่ 2 จังหวะ จึงเป็นการทำงานที่ไม่แตกต่างกันระหว่างสัญญาณไฟจราจรแบบปรกติ (เพราะว่าด้านที่ต่อกับถนนปลายเปิดนั้นจะมีค่าของรถเข้ามาอย่างสม่ำเสมออยู่แล้ว จึงทำให้เวลาในการปล่อยสัญญาณไฟจราจรไม่ได้แตกต่างกันเท่าใดนัก)

แผนที่ที่ 2 ในกรณีที่เลือกถนนที่อยู่กลางแผนที่ (ในการทดสอบนั้นจะเลือกถนน c ซึ่งเป็นถนนที่อยู่กลางแผนที่ โดยเชื่อมต่อกับทางแยก A) จะได้ผลการทำงานทำงานของระบบโดยรวมเป็นดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	Car Rate	0	0	0	0	0
	รถเข้า	0	0	0	0	0
	รถออก	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เปอร์เซ็นต์	15.7681	15.7681	15.7681	15.7681	15.7681
500	Car Rate	-0.7477	-0.6609	-0.0011	0.4044	0.4842
	รถเข้า	92.4923	184.9846	873.4358	1131.6991	1183.7303
	รถออก	466.3890	515.4543	873.9922	929.4567	941.5901
	เปอร์เซ็นต์	1.5999	3.1367	15.8300	23.8541	25.4616
1000	Car Rate	-0.4111	-0.3645	0.0193	0.2250	0.2556
	รถเข้า	184.8000	369.6000	1665.7435	1981.4684	2023.3263
	รถออก	595.9051	734.1985	1646.3883	1756.4518	1767.6782
	เปอร์เซ็นต์	1.6848	2.9563	16.5944	24.7679	26.0036
2000	Car Rate	-0.2342	-0.2076	0.0208	0.1357	0.1538
	รถเข้า	369.4153	738.8307	3250.3588	3697.6993	3745.9041
	รถออก	837.8374	1154.1582	3208.6507	3426.1624	3438.2194
	เปอร์เซ็นต์	1.8017	2.9214	17.5347	26.4338	27.8189
4000	Car Rate	-0.1438	-0.1266	0.0224	0.1134	0.1240
	รถเข้า	738.6461	1477.2923	6419.5895	7142.4688	7191.6429
	รถออก	1313.8516	1983.7902	6329.9420	6688.8596	6695.4755
	เปอร์เซ็นต์	1.6035	2.6532	19.4874	33.5448	35.1870
6000	Car Rate	-0.1138	-0.1021	0.0319	0.1171	0.1259
	รถเข้า	1107.8769	2215.7538	9588.8202	10581.9306	10642.9732
	รถออก	1790.6976	2828.5025	9397.0314	9878.8575	9887.3373
	เปอร์เซ็นต์	1.6467	2.9067	23.6114	43.2256	45.3057
8000	Car Rate	-0.1004	-0.0891	0.0416	0.0936	0.0971
	รถเข้า	1477.1077	2954.2154	12758.0509	13630.3922	13659.9647
	รถออก	2280.7412	3667.4381	12424.9031	12881.5685	12882.5901
	เปอร์เซ็นต์	1.5618	3.0812	29.0538	44.5576	45.6229
10000	Car Rate	-0.0912	-0.0813	0.0467	0.0734	0.0761
	รถเข้า	1846.1538	3692.3077	15925.8969	16602.6230	16630.1934
	รถออก	2758.8844	4505.4269	15458.0796	15868.0529	15868.4530
	เปอร์เซ็นต์	1.7229	2.8801	34.3781	43.4981	44.5080

ตารางที่ 5-18 แสดงผลการทำงานของระบบสัญญาณไฟจราจร
แบบเลือกเส้นทางที่รถติดน้อยที่สุดในแผนที่ที่ 2 (เลือกถนนกลางแผนที่)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนรายละเอียดของถนนนั้น แสดงได้ดังตาราง

เวลาของการทำงาน		ปริมาณรถที่เข้าสู่ระบบ (เปอร์เซ็นต์)				
		0.05	0.1	0.5	1	5
0	$Car\ Rate_{Normal}$	0	0	0	0	0
	$Car\ Rate_{Select}$	0	0	0	0	0
500	$Car\ Rate_{Normal}$	-0.0300	-0.0300	-0.0300	-0.0300	-0.0300
	$Car\ Rate_{Select}$	-0.0258	-0.0184	0.0327	0.0277	0.0277
1000	$Car\ Rate_{Normal}$	-0.0102	-0.0089	-0.0085	-0.0071	-0.0071
	$Car\ Rate_{Select}$	-0.0112	-0.0079	0.0084	0.0174	0.0174
2000	$Car\ Rate_{Normal}$	-0.0074	-0.0074	-0.0075	-0.0075	-0.0075
	$Car\ Rate_{Select}$	-0.0062	-0.0049	0.0075	0.0094	0.0094
4000	$Car\ Rate_{Normal}$	-0.0024	-0.0035	-0.0018	-0.0017	-0.0017
	$Car\ Rate_{Select}$	-0.0024	-0.0025	0.0058	0.0054	0.0054
6000	$Car\ Rate_{Normal}$	-0.0013	-0.0011	0.0013	0.0016	0.0016
	$Car\ Rate_{Select}$	-0.0017	-0.0018	0.0023	0.0036	0.0036
8000	$Car\ Rate_{Normal}$	-0.0018	-0.0018	-0.0018	-0.0018	-0.0018
	$Car\ Rate_{Select}$	-0.0015	-0.0013	8.8038E-4	0.0027	0.0027
10000	$Car\ Rate_{Normal}$	-0.0011	-0.0013	-8.7764E-4	-7.1446E-4	-7.1446E-4
	$Car\ Rate_{Select}$	-0.0011	-0.0011	0.0016	0.0021	0.0021

ตารางที่ 5-19 แสดงค่า Car Rate ของถนนที่เลือก (ถนนกลางแผนที่) ในแผนที่ที่ 2 เปรียบเทียบกับสัญญาณไฟจราจรแบบปรกติ

จากตารางจะเห็นได้ว่าการทำงานของระบบสัญญาณไฟจราจรในแบบนี้นั้นมีการทำงานปล่อยรถออกจากถนนที่เลือกก่อนเสมอ แต่ว่าส่วนที่ผลต่อการทำงานของระบบสัญญาณไฟจราจรในแบบนี้ก็คือปัจจัยรอบด้านของถนนเช่น สัญญาณไฟจราจรที่ทางแยก ปริมาณรถจากถนนอื่นๆ ที่เข้ายังถนนที่สนใจเป็นต้น (ในการทำงานนั้นจะมีการทำงานของสัญญาณไฟจราจรที่มีประสิทธิภาพมากกว่าการทำงานของสัญญาณไฟจราจรแบบปรกติ จะเห็นได้ว่าค่าของ Car Rate นั้นส่วนมากมีค่าเป็น + แสดงว่าปริมาณรถเข้าสู่ถนนนั้นมากกว่ารถออกจากถนน ส่วนค่าที่เป็น – นั้นหมายความว่าปริมาณรถออกจากถนนมากกว่าปริมาณรถที่เข้าที่เข้านั่นเอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.5 ผลการทำงานโดยรวมของระบบ

ผลการทำงานโดยรวมของระบบนั้น สามารถสรุปการทำงานออกมาเป็นตาราง โดยมีเวลาในการทำงานเป็น 10000 วินาที ในแผนที่ B และมีค่าของเปอร์เซ็นต์ของรถที่เข้าสู่ถนนเป็นค่าต่างๆ ดังนี้

ปริมาณรถที่เข้าสู่ถนนปลายเปิดเป็น 0.05% ของปริมาณรถสูงสุดที่ถนนปลายเปิดนั้นรับได้ (ค่าที่น้อยที่สุดที่ใช้ในการทดสอบ) ได้ค่าเป็นดังตาราง

อัลกอริทึมที่ใช้	ปริมาณรถเข้า	ปริมาณรถออก	เปอร์เซ็นต์ของรถในระบบ	Car Rate	ผลการทำงาน
ปรกติ	1846.1538	3057.3222	0.6069	-0.1211	ดี
Best Select	1846.3385	2322.5912	1.3928	-0.0476	ดี
Best Time	1846.3385	1938.7077	1.1304	-0.0092	ดี
Select Road	1846.1538, 1846.1538	2814.4547, 2758.8844	1.5322, 1.7229	-0.0968, -0.0912	ดี, ดี

ตารางที่ 5-20 แสดงผลการทำงานของอัลกอริทึมโดยรวมที่เปอร์เซ็นต์ของรถเป็น 0.05

จากตารางจะเห็นได้ว่าในการทดสอบนั้น การคำนวณสัญญาณไฟจราจรแบบ AI นั้นมีการทำงานที่ดีกว่าการทำงานของระบบสัญญาณไฟจราจรแบบปรกติ (โดยดูจากค่าของปริมาณรถเข้า รถออกจากระบบ และเปอร์เซ็นต์ของรถที่อยู่ในระบบ) ส่วนอัลกอริทึมของ Select Road นั้นมีสองค่าคือ ค่าแรกเป็นการเลือกถนนที่เป็นถนนปลายเปิด ส่วนค่าที่สองเป็นการเลือกถนนที่ยู่ภายในแผนที่ ซึ่งค่าที่ได้ก็มีค่าไม่เท่ากัน และเมื่อเราประเมินความสามารถของการปล่อยรถนั้น โดยเทียบกับการปล่อยสัญญาณไฟจราจรแบบปรกติ เราจะได้ว่าในการประเมินผลการทำงานนั้นจะต้องพิจารณาค่าของ ปริมาณรถเข้าสู่ระบบ (พื้นที่ว่างของถนนมาก → ปริมาณรถเข้าถนนมาก), ปริมาณรถออกจากระบบ (การจัดการสัญญาณไฟจราจร), เปอร์เซ็นต์ของรถที่อยู่ในถนน (ถ้ามีค่าน้อยแสดงว่าการทำงานของสัญญาณไฟจราจรมีประสิทธิภาพมาก)

จากตารางที่แสดงผลการทำงานนั้นจะเห็นได้ว่า เมื่อมีปริมาณรถที่เข้าสู่ถนนนั้นน้อย การทำงานของระบบทุกระบบก็จะสามารถทำงานได้ดี เนื่องจากว่าปริมาณรถน้อยนั่นเองจึงไม่เกิดปัญหารถติด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริมาณรถที่เข้าสู่ถนนปลายเปิดเป็น 0.5% ของปริมาณรถสูงสุดที่ถนนปลายเปิดนั้นรับได้ (ค่าที่ทำให้ระบบอยู่ในระดับที่ Critical พอดี) ได้ดังนี้

อัลกอริทึมที่ใช้	ปริมาณรถเข้า	ปริมาณรถออก	เปอร์เซ็นต์ของรถในระบบ	Car Rate	ผลการทำงาน
ปรกติ	15925.8969	15541.8969	35.5988	0.0384	แย่
Best Select	18461.5385	18449.8311	10.4620	0.0011	ดีที่สุด
Best Time	18382.5627	18461.5385	11.0171	-0.0065	ดี
Select Road	18461.5385, 15925.8969	18513.6647, 15458.0796	13.5312, 34.3781	-0.0052, 0.0467	ดี, พอใช้

ตารางที่ 5-21 แสดงผลการทำงานของอัลกอริทึมโดยรวมที่เปอร์เซ็นต์ของรถเป็น 0.5

จากตารางแสดงผลการทำงานของระบบสัญญาณไฟจราจรแบบต่างๆ นั้นเราจะได้ว่าที่การทำงานของ Best Select นั้นทำงานได้ดีที่สุด (เพราะว่ามีเปอร์เซ็นต์ของรถในถนนน้อยที่สุด ในขณะที่ปริมาณรถเข้าสู่ระบบมากที่สุด) ส่วนการทำงานของ Best Time นั้นมีผลการทดสอบที่รองลงมา คือมีการปล่อยที่ใกล้เคียงกับ Best Select มาก ส่วนการทำงานของ Select Road นั้นจะมีผลการทำงานที่ดีเมื่อเลือกถนนที่ปลายเปิด แต่จะทำงานได้ใกล้เคียงกับปรกติเมื่อเลือกถนนที่อยู่ในแผนที่

ปริมาณรถที่เข้าสู่ถนนปลายเปิดเป็น 5% ของปริมาณรถสูงสุดที่ถนนปลายเปิดนั้นรับได้ (ปริมาณรถเข้าสู่ระบบมีมากที่สุด เมื่อมีค่ามากก็แสดงว่าถนนจะเข้าสู่สภาวะการติดอย่างรวดเร็ว) มีผลการทำงานของอัลกอริทึมต่างๆ แสดงได้ดังตาราง

อัลกอริทึมที่ใช้	ปริมาณรถเข้า	ปริมาณรถออก	เปอร์เซ็นต์ของรถในระบบ	Car Rate	ผลการทำงาน
ปรกติ	16568.5267	15937.8560	43.9114	0.0633	ปานกลาง
Best Select	10587.3048	9578.7311	57.5592	0.1008	แย่
Best Time	18316.1888	17489.3656	44.4580	0.0826	ปานกลาง
Select Road	18771.7224, 16630.1934	18614.7026, 15868.4530	22.5919, 44.5080	0.0157, 0.0761	ดี, ปานกลาง

ตารางที่ 5-22 แสดงผลการทำงานของอัลกอริทึมโดยรวมที่เปอร์เซ็นต์ของรถเป็น 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางจะเห็นได้ว่าการทำงานของ Select Road นั้นทำงานได้ดี เมื่อเลือกที่ถนนปลายเปิด แต่ปัจจัยที่มีผลต่อค่าดังกล่าวก็คือ ถนนเส้นนั้นผ่านสี่แยกเดียว (สี่แยก B) แล้วก็ออกจากระบบเลย เป็นสาเหตุที่ทำให้รถเข้าและออกจากระบบได้มากนั่นเอง แต่การทำงานของ Best Select นั้นจะมีการทำงานที่แย่ที่สุด เนื่องมาจากว่า การปล่อยรถในแต่ละทางนั้นการคำนวณสัญญาณไฟจราจรจะต้องปล่อยรถได้มากที่สุด แต่เมื่อถนนเต็มหมดก็จะปล่อยสัญญาณไฟแต่ละจังหวะแบบสั้นๆ เป็นผลให้รถวิ่งได้น้อย อีกทั้งยังเป็นการเปลืองการทำงานของคอมพิวเตอร์ด้วยในการคำนวณหาสัญญาณไฟจราจรที่ดีที่สุด หรือเราอาจจะสรุปง่ายๆ ได้ว่าเมื่อปริมาณรถมีมากก็จะจัดการระบบสัญญาณไฟจราจรได้อย่างยากลำบาก แต่ถ้ามีปริมาณรถในระดับ Critical แล้วละก็จะสามารถจัดการจราจรได้อย่างมีประสิทธิภาพมากนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุป

6.1 สรุปโครงงาน

ในการสรุปการทำงานนั้นเป็นส่วนสำคัญ เพราะเป็นส่วนที่แสดงให้เห็นว่ามีความสัมพันธ์กันของการทดลองว่าเป็นอย่างไร

ในโครงงานชิ้นนี้เป็นการทำงานของระบบสัญญาณไฟจราจร มีส่วนของการทำงานเป็นหลายส่วนด้วยกัน ซึ่งเราจะแบ่งหัวข้อของการสรุปออกเป็นส่วนย่อยๆ ทีละส่วนก่อนแล้วจึงสรุปการทำงานโดยรวมทีหลัง

6.1.1 สรุปการทำงานในส่วนของโมเดลที่ใช้ในโปรแกรม

เป็นส่วนสำคัญของโปรแกรมเลยก็ว่าได้ เพราะว่าทุกส่วนของโปรแกรมจะต้องมาเชื่อมต่อกับส่วนนี้ เราสรุปว่า "*โมเดลที่ใช้ในโปรแกรมนั้นสามารถทำงานได้ใกล้เคียงกับความจริงมาก ส่วนรายละเอียดปลีกย่อยต่างๆ เป็นส่วนที่เกิดขึ้นในทางปฏิบัติทั้งสิ้น ดังนั้นเราจึงจำเป็นจะต้องนำข้อมูลจริงมาใช้ในการคำนวณด้วย*"

รายละเอียดปลีกย่อยดังกล่าวเช่น ความเร่งของรถแต่ละคัน, ปริมาณ Car Rate ที่วิ่งได้จริงจะไม่คงที่เสมอ, อุบัติเหตุ เป็นต้น

6.1.2 สรุปการทำงานในส่วนของ Editor

การทำงานของ Editor ที่ใช้ในโปรแกรม ยังมีการทำงานที่จัดว่าพอใช้ได้ แต่ว่าผู้ใช้ยังต้องอ่านคู่มืออยู่สักพักหนึ่งก่อน เราสรุปว่า "*การทำงานในส่วนของ Editor นั้นมีการทำงานที่จัดว่าง่ายต่อการใช้แต่ยังต้องปรับปรุงการทำงานในบางส่วนเพื่อให้โปรแกรมใช้งานได้มากยิ่งขึ้น*"

6.1.3 สรุปการทำงานในส่วนของการจำลองการทำงานของสัญญาณไฟ

การทำงานในส่วนของสัญญาณเป็นส่วนที่ใช้ทดสอบโมเดลที่ได้ออกแบบไว้ ซึ่งจากการจำลองการทำงานของสัญญาณไฟ มีการทำงานในหลายๆ ส่วน เราสรุปการทำงานของแต่ละส่วนออกเป็นตารางคร่าวๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของอัลกอริทึม	เวลาที่ใช้ในการทำงานในระบบขนาด			ประสิทธิภาพในการทำงานในระบบขนาด			สรุปโดยรวม
	เล็ก	กลาง	ใหญ่	เล็ก	กลาง	ใหญ่	
Normal Mode (ทำงานแบบอัตโนมัติ)	เร็วมาก	เร็ว	เร็ว	ดี	ปานกลาง	แย่	ดีในระบบขนาดเล็ก
Best Time Mode (ทำงานแบบคำนวณเวลาอย่างเดียว)	เร็ว	เร็ว	ปานกลาง	ดี	ดี	ดีมาก	ดีในระบบทุกขนาด
Best Select Mode (ทำงานแบบคำนวณเวลาและสัญญาณไฟจราจร)	ปานกลาง	ช้า	ช้ามาก	ดี	ดี	ดีมาก	ดีในระบบทุกขนาด
Select Road Mode (ทำงานแบบคำนวณเวลาและสัญญาณไฟจราจร)	ปานกลาง	ช้า	ช้ามาก	ปานกลาง	ปานกลาง	ปานกลาง	ดีเฉพาะในถนนที่เลือก (จะมีผลต่อระบบในกรณีที่เลือกถนนปลายเปิด)

ตารางที่ 6-1 แสดงผลสรุปของอัลกอริทึมที่ได้ออกแบบมาเปรียบเทียบกัน

จากสรุปการทำงานในตารางจะเห็นได้ว่ามีการใช้ระบบขนาดต่างๆ มาทดสอบ ไม่ว่าจะเป็นระบบขนาดเล็ก ระบบขนาดกลาง และระบบขนาดใหญ่ ซึ่งระบบขนาดต่างๆ นี้เปรียบเสมือน Benchmark ในการทดสอบการทำงานของทั้งสองโหมดการทำงาน จะเห็นได้ว่าในระบบขนาดเล็กนั้นการทำงานด้วยระบบอัตโนมัติจะมีการทำงานที่มีประสิทธิภาพมากกว่าการทำงานด้วยโหมดการทำงานแบบการคำนวณสัญญาณไฟจราจร (ในโหมดของการทำงานแบบ AI) ส่วนการทำงานแบบ AI ทั้งสามแบบนั้นต่างก็มีพื้นฐานมาจากการหาเวลาที่ใช้ให้คุ้มค่าเนื่องจากว่าในระบบการทำงานแบบปรกตินั้นไม่มีการสนใจปริมาณรถในการปล่อยสัญญาณไฟจราจรเลย ซึ่งวัตถุประสงค์ของการทำงานของ AI แต่ละแบบนั้นเราสามารถสรุปการทำงานออกเป็นส่วนๆ ได้ดังนี้

- ในโหมดการทำงานแบบอัตโนมัติ "เป็นการทำงานที่ถูกต้องตามความเป็นจริง อีกทั้งผู้ใช้สามารถเปลี่ยนสัญญาณไฟจราจรเองได้ด้วย มีการทำงานที่ถูกต้องตามวัตถุประสงค์"
- ในโหมดการทำงานแบบหาเวลาที่ดีที่สุด (Best Time Mode) สรุปได้ว่า "เป็นการทำงานที่ใช้เวลาการปล่อยสัญญาณไฟจราจรได้คุ้มค่าที่สุด"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในโหมดการทำงานแบบเลือกสัญญาณไฟจราจรที่ดีที่สุด (Best Select Mode) สรุปได้ว่า "เป็นการทำงานที่ปล่อยสัญญาณไฟจราจรได้ดีที่สุด ในโครงงานที่ได้ออกแบบมา"
- ในโหมดการทำงานแบบเลือกถนนที่ต้องการให้รถติดน้อยที่สุด (Select Road Mode) สรุปได้ว่า "การเลือกถนนที่สนใจนั้นส่งผลต่อการทำงานโดยรวมของระบบ ซึ่งอาจจะทำงานได้ดีมาก หรือว่าเทียบเท่ากับสัญญาณไฟจราจรแบบปรกติก็ได้ ขึ้นอยู่กับปัจจัยหลายอย่าง"

6.1.4 สรุปการทำงานโดยรวมของระบบ

จากระบบที่ได้ออกแบบและพัฒนานั้น สามารถสรุปได้ว่า "*การทำงานของระบบสัญญาณไฟจราจรนั้นมีความสอดคล้องใกล้เคียงกับความเป็นจริงของระบบสัญญาณไฟจราจรที่ใช้งานจริงและส่วนของอัลกอริทึมที่ใช้ในการหาสัญญาณไฟจราจรนั้นมีประสิทธิภาพของการทำงานดีในระดับหนึ่ง*"

6.2 แนวทางการพัฒนาต่อ

แนวทางการพัฒนาโปรแกรมสามารถแบ่งการพัฒนาออกได้เป็น 3 ส่วนคือ

1. ส่วนของ Editor สามารถขยายการทำงาน
   - สามารถขยาย Object ของถนนให้มากกว่านี้ได้อีก โดยอาจจะเพิ่มทางโค้ง หรือถนนที่ปรับองศาได้อิสระ
   - เพิ่ม Attribute ของ Object ต่างๆ ภายในโปรแกรม ให้เกิดสภาพใกล้เคียงความเป็นจริงมากที่สุด (ในโปรแกรมมีการสมมติค่าไว้หลายอย่างทำให้โปรแกรมที่ใช้ได้ยังมีความสามารถไม่เพียงพอต่อการทำงานจริง)
   - ใช้ระบบ Image Processing ในการช่วยหาการต่อของถนน เพื่อให้ทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น
2. ส่วนของการจำลองการทำงานของระบบสัญญาณไฟจราจร สามารถขยายการทำงานได้เป็น
   - ส่วนของ AI สามารถหาอัลกอริทึมที่ช่วยจัดการระบบได้ดีกว่าที่ออกแบบเอาไว้ในโครงงาน
   - เพิ่มความสามารถในการทำงานของ Run โปรแกรมให้มีความสามารถของการทำงานได้ใกล้เคียงกับความเป็นจริงมากที่สุด ซึ่งในโปรแกรมได้มีการกำหนดค่าของสัญญาณไฟของทางแยกได้เองด้วย (อาจจะเรียกได้ว่าเป็นการ Manual control traffic) แต่สามารถกำหนดการทำงานของทางแยกได้เพียงทางแยกเดียว ควรจะเพิ่มการทำงานได้หลายทางแยก
3. ส่วนของการเพิ่มความสามารถในการทำงานของโปรแกรม เราจะสามารถเพิ่มการทำงานของโปรแกรมให้มากยิ่งขึ้นได้โดย เพิ่มการทำงานผ่านทาง Network เพื่อให้สามารถคำนวณและให้ประโยชน์แก่ระบบสัญญาณไฟจราจรได้จริง, ติดตั้งกล้องหรือ Sensor ช่วยหาปริมาณรถภายในถนน, ติดตั้งระบบตรวจจับความเร็วของรถ (ในโครงงานเรียกว่า Car Rate) เพื่อช่วยในการประเมินความสามารถในการทำงานของระบบ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ขยายการทำงานโดยเชื่อมโยงกับเครือข่ายที่มีความเร็วสูง เพื่อช่วยรายงานสภาพของถนนในโปรแกรม และช่วยรายงานผลการทำงานของระบบสัญญาณไฟจราจรให้บุคคลทั่วไปได้รับรู้
5. อาศัยระบบฐานข้อมูลช่วยเหลือในการทำงานของระบบ โดยขยายระบบให้มีขนาดใหญ่มากๆ และอาศัยระบบฐานข้อมูลในการเก็บค่าต่างๆ ที่ได้คำนวณออกไปแล้วลงสู่ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ภาษา JAVA

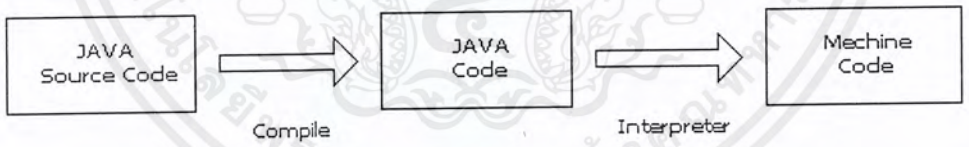
ภาษา JAVA เป็นภาษาที่ได้รับความนิยมอย่างมาก โดยมีจุดเด่นที่สามารถนำไปใช้ได้กับเครื่องคอมพิวเตอร์เครื่องอื่นได้โดยไม่ต้องทำการคอมไพล์โปรแกรมใหม่ สามารถนำไปใช้งานได้โดยทันที จึงเป็นจุดเด่นที่ใช้งานกันมากบน Internet และระบบที่มีเครือข่ายใหญ่ๆ

ภาษา JAVA ถูกออกแบบขึ้นโดยมีจุดมุ่งหมายดังนี้

1. เป็นภาษาที่ง่าย มีกลไกของภาษาจำนวนไม่มาก และไม่ซับซ้อน โดยมีไวยากรณ์ต่างๆ คล้ายกับภาษา C++ มากทำให้ง่ายต่อการอ่านและทำความเข้าใจกับโค้ดของโปรแกรม
2. โปรแกรมที่สร้างขึ้นจะต้องมีความผิดพลาดที่ไม่เคยคาดคิดมาก่อนน้อยที่สุดและโปรแกรมจะต้องไม่ล้มเหลวลงง่ายๆ ด้วยความผิดปกติเพียงเล็กน้อย
3. โปรแกรมที่สร้างขึ้นมักจะถูกส่งผ่านไปยังระบบเครือข่ายๆปทำงานบนเครื่องคอมพิวเตอร์ของผู้อื่น ดังนั้นมันจะต้องไม่ก่อให้เกิดอันตรายต่อเครื่องหรือระบบของผู้ใช้ (มี Security มาก)

เนื่องจากการทำงานของภาษา JAVA จะทำงานบนเครื่องคอมพิวเตอร์ได้หลายๆ เครื่อง ดังนั้นโปรแกรมจะต้องอาศัยการทำงานโดยใช้ Interpreter ช่วยในการทำงานซึ่งอาจจะเรียกว่า Java's Virtual Machine (JVM) ช่วยในการทำงาน

ลักษณะการทำงานของภาษา JAVA จะมีลักษณะดังรูป



รูปที่ ก -1 แสดงลักษณะโดยรวมของภาษา JAVA

จากรูป เราจะเห็นกลไกในการทำงานของภาษา JAVA ว่ามีการทำงานโดยการคอมไพล์จากโปรแกรมมิ่งโค้ดไปเป็นโค้ดของ JAVA ซึ่งในภาษา JAVA จะเรียกว่า คลาสไฟล์ (Class File) ซึ่งจะมีนามสกุลเป็น .class เมื่อใดที่ต้องการจะ Run โปรแกรมก็จะต้องอาศัย Java Interpreter ในการทำงาน

รายละเอียดของภาษา JAVA

ในปัจจุบันได้มีผู้ผลิตโปรแกรมภาษา JAVA ออกมาเป็นจำนวนมาก เช่น Java Developer Kit (JDK), Jbuilder, Visual Café, Visual J++ เป็นต้น แต่เนื่องจากว่าภาษา JAVA นั้นได้รับการพัฒนาจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sun Microsystem และผู้ผลิตรายอื่นๆ ได้ทำงานใช้งานจาก Sun อีกทีหนึ่ง ดังนั้นในโครงงานนี้จึงได้เลือกใช้ Java Developer Kit ในการทำงานเป็นหลัก

โปรแกรม Java Developer Kit สามารถ download ได้จาก http://www.javasoft.com โดยจะมีรายละเอียดดังนี้

1. ส่วนของคอมไพล์เลอร์นั้นจะใช้โปรแกรม javac.exe ในการคอมไพล์
2. ส่วนของการ run โปรแกรมนั้นจะใช้โปรแกรม java.exe ในการ run โปรแกรม ซึ่งจะต้องมี Java Runtime Environment (JRE) ร่วมด้วยในการทำงาน

ก.1 ขั้นตอนการติดตั้ง

ขั้นตอนการติดตั้ง Java Developer Kit หลังจากที่ได้ download มาแล้วทำการติดตั้งลงบนเครื่องคอมพิวเตอร์ สมมติว่าที่ C:\jdk1.3.1 จากนั้นทำการอ้างอิงค่าในส่วนของ Compiler และ Interpreter ในไฟล์ AUTOEXEC.BAT ที่ root directory ของเครื่อง (สำหรับระบบปฏิบัติการบน window) โดยเพิ่มคำสั่งดังนี้

SET PATH=C:\JDK1.3.1\BIN

ลงไปในไฟล์ AUTOEXEC.BAT เป็นการอ้างอิง Path ของเครื่อง ซึ่งหมายความว่าสามารถที่จะเรียกใช้โปรแกรมที่ใดก็ได้ในเครื่องผ่านทาง Dos ของ window จากนั้นทำการสร้าง Directory ที่ทำงาน java บนเครื่องเช่น C:\My Documents\Java ก็จะต้อง set ค่าในไฟล์ AUTOEXEC.BAT เป็นดังนี้

SET CLASSPATH=.;C:\MYDOCU~1\JAVA;C:\JDK1.3.1\LIB\CLASSES.ZIP

เมื่อทำการแก้ไขไฟล์ AUTOEXEC.BAT แล้วก็ทำการ restart เครื่อง ก็จะพร้อมใช้งานแล้ว

ก.2 สาเหตุการใช้ภาษา JAVA

สาเหตุที่ใช้ภาษา JAVA ในโครงงาน มีสาเหตุดังนี้

1. เนื่องจากภาษา JAVA สามารถทำงานได้หลายระบบปฏิบัติการ และระบบคอมพิวเตอร์ที่ใช้งานส่วนใหญ่มักจะใช้ Linux หรือไม่ก็ UNIX ในการทำงาน เพราะว่าเสถียร (Stable) แต่โครงงานที่ทำการพัฒนานั้นพัฒนาบนระบบปฏิบัติการ Window Me ดังนั้นจึงต้องอาศัย JAVA ในการทำงาน
2. ภาษา JAVA กำลังได้รับความนิยมอย่างมากในปัจจุบันอีกทั้งยังมีการพัฒนาอย่างต่อเนื่อง ทำให้สามารถใช้งานได้อย่างยาวนาน
3. ภาษา JAVA มีความเสถียรภาพมาก สามารถทำงานได้อย่างดี (เพราะว่าโปรแกรมถูกออกแบบมาเป็นอย่างดีนั่นเอง)
4. เมื่อทำงานบนระบบปฏิบัติการเช่น UNIX หรือ Linux สามารถเชื่อมต่อระบบคอมพิวเตอร์ให้ใหญ่ขึ้นได้ เพื่อช่วยในการคำนวนระบบที่มีขนาดใหญ่ได้อย่างถูกต้องและแม่นยำยิ่งขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่การใช้ภาษา JAVA ก็มีข้อเสียหลายอย่าง คือ สิ้นเปลืองทรัพยากรของเครื่องคอมพิวเตอร์อย่างมาก เพราะว่าจะต้องแปลงโค้ดจาก JAVA code มาเป็น Machine code อีกทั้งยังต้องสำรองหน่วยความจำเอาไว้ในการแปลงโค้ดดังกล่าวด้วย

ซึ่งจากข้อเสียดังกล่าว ในอนาคตจะมีการพัฒนาและปรับปรุงอุปกรณ์ฮาร์ดแวร์แล้วก็จะแก้ปัญหาดังกล่าวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ระบบการจราจร (Traffic System)

ระบบการจราจรในทางปฏิบัตินั้น มีการทำงานที่ซับซ้อนและมีปัจจัยหลายอย่างในการทำงาน ไม่ว่าจะเป็นปัจจัยทางด้านกายภาพ หรือปัจจัยทางด้านสัญญาณไฟ หรืออาจจะเป็นปัจจัยทางด้านอากาศ ซึ่งปัจจัยดังกล่าวนั้นบางอย่างสามารถควบคุมได้ บางอย่างก็ไม่สามารถควบคุมได้

ข.1 การพัฒนาของทางแยก (Cross Road Generation)

การพัฒนาของทางแยก (Cross Road Generation) เป็นลำดับขั้นของการพัฒนาของทางแยกซึ่งมีลำดับขั้นของการพัฒนาดังรูปที่ ข-1

ซึ่งจากรูปจะเห็นลำดับขั้นของการพัฒนาของทางแยก ซึ่งมีการพัฒนาดังนี้

ขั้นที่ 1 เป็นการจัดการทางแยกแบบที่แย่ที่สุด กล่าวคือไม่มีการจัดการทางแยกเลย ปล่อยให้รถวิ่งตามปรกติ (ไม่มีการเปิดสัญญาณไฟบอกด้วยว่าข้างหน้ามีทางแยกหรือไม่)

ขั้นที่ 2 เป็นการจัดการทางแยกในลำดับขั้นต่อมา คือมีการติดตั้งสัญญาณไฟจราจรเอาไว้ แต่ไม่มีการเปิดใช้ เปิดใช้แต่เพียงสัญญาณไฟเหลืองเท่านั้น (เป็นสัญญาณไฟที่บอกให้คนขับระวังข้างหน้า เพราะว่ามีทางแยกที่เปิดใช้การแล้ว) หรืออาจจะเป็นป้ายหยุดรถที่บอกเตือนให้คนขับระวังทางข้างหน้า

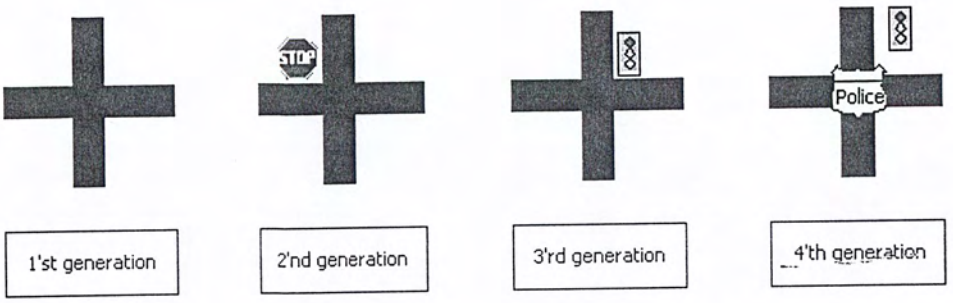
ขั้นที่ 3 เป็นการจัดการทางแยก แบบที่มีการติดตั้งและกำหนดสัญญาณไฟจราจรเอาไว้ แต่เป็นการติดตั้งสัญญาณไฟแบบอัตโนมัติ กล่าวคือเป็นการติดตั้งสัญญาณไฟที่มีการปล่อยสัญญาณไฟแบบตั้งเวลาเอาไว้ว่าสัญญาณนี้ปล่อยได้เป็นเวลาเท่าใด เหมือนกับในโครงงานในหัวข้อที่ 3.2.3.15 (Normal Mode)

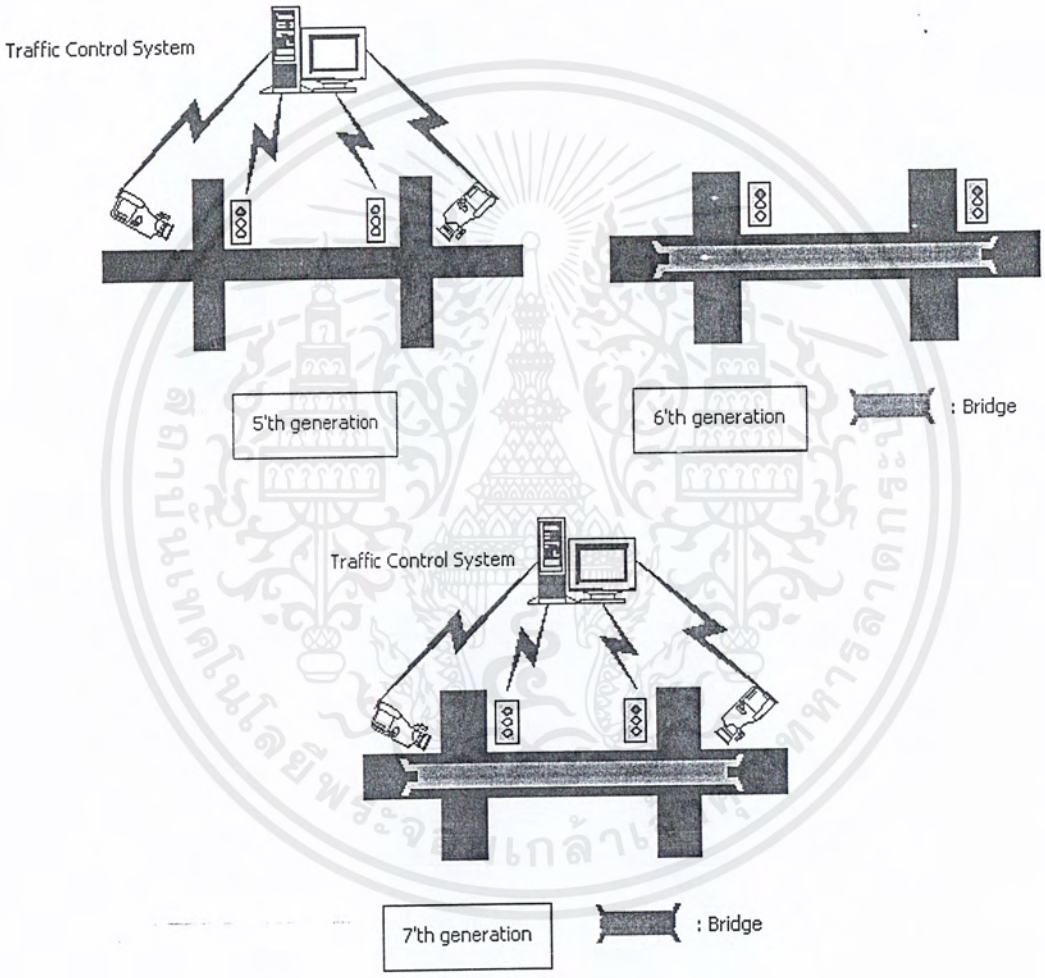
ขั้นที่ 4 เป็นการใช้ตำรวจจราจรในการปล่อยรถที่ทางแยก โดยดูว่าทางใดปริมาณรถมากก็ปล่อยทางนั้นไปก่อน เหมือนกับในโครงงานหัวข้อที่ 3.2.6.1 (Best Select Mode)

ขั้นที่ 5 เป็นการเชื่อมต่อถนนที่มีการติดขัดของรถหลายๆ ถนนเข้าด้วยกัน โดยพิจารณาว่าจะปล่อยรถทางใดให้ได้มากที่สุด ซึ่งมีการเชื่อมต่อกันของทางแยกหลายๆ ทางแยก เพื่อช่วยในการทำงาน ซึ่งในทางปฏิบัติจะใช้ตำรวจจราจรหลายๆ คนควบคุมทางแยกหลายๆ ทางแยก หรืออาจจะใช้อุปกรณ์ช่วยเหลือในการทำงาน (ในนั้นกรุงเทพมหานครมีการใช้ ซึ่งระบบดังกล่าวที่นำมาใช้นั้นจะเรียกว่าเป็นระบบ Area Traffic Control; ATC รายละเอียดการทำงานจะกล่าวในหัวข้อที่ ข.2)

ขั้นที่ 6 เป็นการสร้างสะพานข้ามแยก หรืออุโมงค์ลอดทางแยก เพราะว่าการทำงานของระบบในขั้นที่ 6 นั้นไม่สามารถทำงานได้แล้ว หรืออาจจะทำงานได้ไม่มีประสิทธิภาพนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ ข-1 แสดงลำดับขั้นของการพัฒนาของทางแยก

ขั้นที่ 7 เป็นการผสมผสานการทำงานระหว่าง การสร้างสะพานข้ามแยก หรืออุโมงค์ลอดทางแยก และการเชื่อมต่อทางแยกด้านล่างด้วย โดยใช้ระบบคอมพิวเตอร์มาช่วยในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางปฏิบัตินั้นการเลื่อนลำดับขั้นของทางแยกนั้นไม่จำเป็นจะต้องเลื่อนตามปรกติ อาจจะเลื่อนข้ามขั้นก็ได้ ทั้งนี้ขึ้นอยู่กับปัจจัยหลายๆ อย่าง ไม่ว่าจะเป็นขนาดของทางแยก ถนนที่เชื่อมต่อหรือเกี่ยวข้อง สภาพภูมิประเทศ สภาพภูมิอากาศ ราคาของการก่อสร้าง ความจำเป็นของถนน หรืออาจจะเป็นปัจจัยอื่นๆ ที่มีผลเกี่ยวข้อง แต่การพัฒนาของทางแยกโดยทั่วไปแล้วสามารถพัฒนาได้สูงสุดถึงขั้นที่ 5 เพราะว่าทุกทางแยกจะต้องมีการสร้างสัญญาณไฟจราจรอยู่แล้ว และเมื่อจัดการระบบสัญญาณไฟจราจรให้ถูกต้องก็จะสามารถกระจายปัญหาดังกล่าวออกไปได้

ข.2 Area Traffic Control; ATC

Area Traffic Control หรือ ATC เป็นการนำระบบคอมพิวเตอร์มาใช้ในการควบคุมสัญญาณไฟจราจรที่ทางแยก เพื่อให้ปล่อยรถออกจากถนนที่สนใจได้ดีที่สุด

ในเมืองไทยเองได้มีการใช้ระบบ ATC มาใช้ด้วย โดยสำนักงานคณะกรรมการจัดการระบบการจราจรทางบก สำนักนายกรัฐมนตรี (สจร.) ได้ให้คำนิยามเอาไว้ว่า "เป็นโครงการที่นำคอมพิวเตอร์ระบบ SCOOT มาใช้ควบคุมสัญญาณไฟจราจรที่บริเวณทางแยกต่างๆในพื้นที่ ครอบคลุมให้ทำงานประสานกันเพื่อเพิ่มประสิทธิภาพในการควบคุมการจราจร บรรเทามลภาวะทางอากาศ อันเกิดจาก สภาพการจราจรที่หยุดนิ่งไม่เคลื่อนตัวและลดปัญหาด้านบุคลากรในพื้นที่ ทำให้เจ้าหน้าที่ตำรวจได้มีโอกาสปฏิบัติ งานอำนวยความสะดวกแก่ประชาชน และดูแลรักษาการใช้ยวดยานและท้องถนนให้เป็นไปตามกฎระเบียบอันจะช่วยส่งเสริมการคลี่คลายปัญหาการจราจรติดขัดให้เป็นไปอย่างมีประสิทธิภาพยิ่งขึ้น"

ในหัวข้อนี้จะอธิบายถึงส่วนที่ใช้ในเมืองไทยเท่านั้น (ซึ่งใช้ระบบ ATC ของบริษัท Peek Traffic co, ltd.) มีขั้นตอนการทำงานดังนี้

1. ในแต่ละทางแยกจะมีเครื่องตรวจนับรถยนต์และสภาพจราจรเพื่อส่งข้อมูลไปยังคอมพิวเตอร์ที่ศูนย์ควบคุมฯ
2. คอมพิวเตอร์ที่ศูนย์ควบคุมฯ จะประเมินข้อมูลสภาพจราจรที่ได้ประกอบกับข้อมูลพื้นฐานอื่นๆของแต่ละทางแยกนั้น เช่น ความยาวถนน, จำนวนช่องเดินรถ, ลักษณะการเดินรถ(ทางเดียว-สองทาง) ,จุดเลี้ยวในทางแยก, ความสามารถในการรองรับปริมาณรถยนต์ ฯลฯ
3. เมื่อประเมินผลแล้ว คอมพิวเตอร์จะสั่งการควบคุมการทำงานของสัญญาณไฟจราจรที่ทุกๆ ทางแยก ทั้งในจุดที่มีปัญหาการจราจรติดขัด, จุดที่การจราจรเบาบางและในจุดที่มีเหตุฉุกเฉิน ให้ทำงานประสานกันอย่างต่อเนื่องทุกจุด โดยมุ่งให้ทุกพื้นที่ผิวจราจร มีการใช้ประโยชน์ได้อย่างสูงสุด
4. ขั้นตอนดังกล่าว ระบบคอมพิวเตอร์จะทำหน้าที่สั่งการ และควบคุมการปรับเปลี่ยนสัญญาณไฟจราจรของทางแยกให้สอดคล้องกันตามสภาพจริง ทั้งในสถานการณ์ปกติ และสถานการณ์พิเศษ (เช่น อุบัติเหตุบนพื้นผิวจราจร เส้นทางฉุกเฉินสำหรับรถดับเพลิง รถพยาบาล เป็นต้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเมืองไทยนั้นมีการติดตั้งระบบ ATC นี้ที่ทางแยกต่างๆ หลายทางแยกซึ่งมีรายละเอียดดังนี้ (หมายเหตุ การติดตั้งนี้อ้างอิงข้อมูล ณ. วันที่ 8 สิงหาคม พ.ศ. 2544)

ลำดับที่	รายชื่อของทางแยก
1	แยกดินแดง
2	อนุสาวรีย์ชัยสมรภูมิ
3	ร.พ.พญาไท2
4	แยกสะพานควาย
5	สะพานลอยลาดพร้าว
6	รัชดา – ลาดพร้าว
7	แยกโชคชัย4
8	สามแยกเกษตร
9	อนุสาวรีย์พิทักษ์รัฐธรรมนูญ
10	แยกหลักสี่
11	แยกบางเขน
12	แยกประชานุกูล
13	แยกประชาชื่น
14	แยกบางโพ
15	แยกสะพานแดง
16	แยกเกียกกาย
17	แยกประดิพัทธ
18	แยกสุทธิสาร
19	แยกบางกะปิ
20	แยกลำสาลี
21	แยกพัฒนาการ
22	แยกศรีนุช
23	แยกบางนา
24	แยกสุขุมวิท 71
25	แยกสุขุมวิท – เอกมัย
26	แยกคลองตัน
27	แยกรามคำแหง – พระราม 9
28	แยกมหาดไทย
29	แยกอโศก – เพชร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

30	แยกใต้ด่วนเพชร
31	แยกประตูน้ำ
32	แยกราชประสงค์
33	แยกปทุมวัน
34	แยกพงษ์พระราม
35	แยกยมราช
36	แยกศรีอยุธยา
37	แยกตึกชัย
38	แยกพานิช
39	พระรูป ร.5
40	แยกซังฮี้
41	แยกบางพลัด
42	แยกสิรินธร
43	แยกบรมราชินี
44	แยกพระปิ่นเกล้า
45	แยกหลักเมือง
46	แยกพระนั่งเกล้า
47	แยก S.A.B
48	แยกหัวลำโพง
49	แยกสามย่าน
50	แยกศาลาแดง
51	รัชดา – พระราม 4
52	สีลม – เจริญกรุง
53	สาทร
54	วงเวียนใหญ่
55	แยกถนนตก
56	แยกมไหสวรรค์
57	แยกท่าพระ
58	แยกไฟฉาย
59	เพชรเกษม – บางแค
60	ด่วนสุขสวัสดิ์

ตารางที่ ข-1 แสดงรายชื่อทางแยกที่มีการติดตั้งระบบ ATC แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข-2 แสดงแผนที่ของทางแยกที่มีการติดตั้งระบบ ATC ของกรุงเทพมหานคร

ข.3 สภาพระบบการจราจรของกรุงเทพฯ ในปัจจุบัน

สภาพระบบการจราจรของกรุงเทพในปัจจุบันนั้น มีการใช้งานระบบต่างๆ อย่างมากมายดังที่ได้กล่าวแล้วในหัวข้อข้างต้น แต่ในบางครั้งเราจะรู้สึกว่าเหมือนไม่มีความแตกต่างกันเลยระหว่างระบบสัญญาณไฟจราจรที่มีการเชื่อมต่อ กับระบบแบบปรกติ มีสาเหตุมาจากว่า

1. ปริมาณรถในกรุงเทพมหานครเองนั้นมีปริมาณมาก
2. การขับขี่ยานพาหนะของคนกรุงเทพยังขาดระเบียบวินัย
3. สภาพถนนในกรุงเทพไม่มีการจัดการวางผังเมืองอย่างถูกต้องมาตั้งแต่แรก
4. สภาพของถนนเอง บางส่วนของถนนมีการชำรุด เสียหายทำให้ความเร็วที่ใช้ได้ในถนนนั้นลดลงไป หรือบางทีมีการก่อสร้างเป็นต้น
5. สภาพดินฟ้าอากาศ เช่น เมื่อมีฝนตกลงมา ก็จะทำให้ถนนทางเส้นไม่สามารถใช้การได้จึงจำเป็นจะต้องมีการรวมรถมายังถนนสายหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางสำนักงานจัดการระบบขนส่งทางบกเองได้ทำการแก้ไขและปรับปรุงการทำงานในหลายส่วนแล้ว แต่บางสาเหตุก็จำเป็นจะต้องอาศัยความร่วมมือของหลายๆ ฝ่ายด้วยกัน เพื่อช่วยในการแก้ปัญหาการจราจร ซึ่งเป็นปัญหาเรื้อรังของกรุงเทพมหานครมาอย่างยาวนาน

ข.4 อนาคตของการจราจรในกรุงเทพมหานคร

สำหรับอนาคตของการทำงานของระบบสัญญาณไฟจราจรในกรุงเทพมหานครนั้น จะมีการทำทั้งในส่วนของการจัดสร้างถนน และการติดตั้งระบบ ATC ควบคู่กันไป

เฉพาะในส่วนของการติดตั้งระบบ ATC นั้น มีการแบ่งการทำงานออกเป็น 3 ระยะด้วยกัน ด้วยกัน ดังนี้

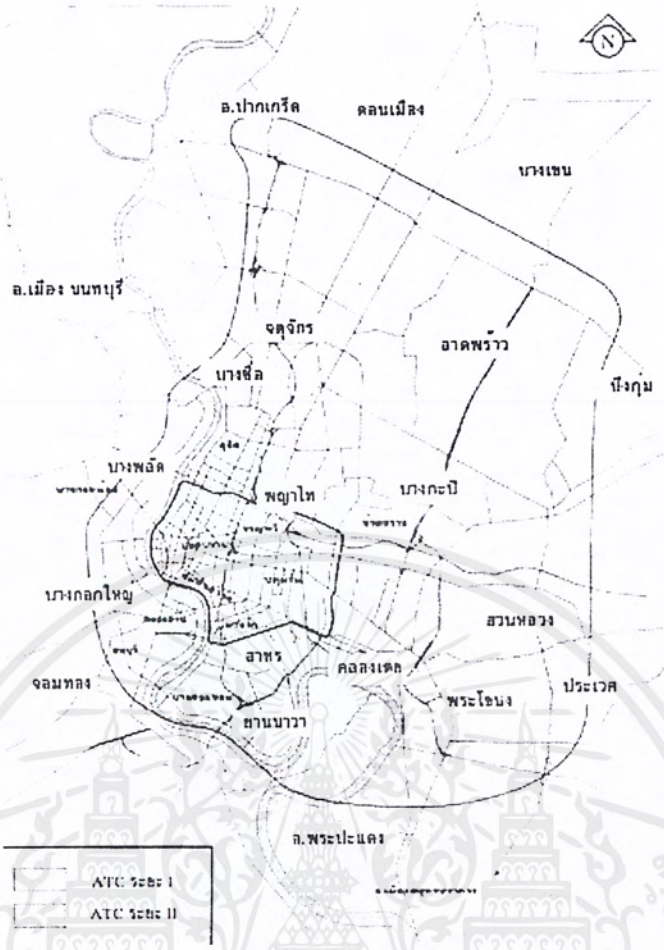
ระยะที่ 1 ครอบคลุมพื้นที่ 31 ตารางกิโลเมตรในเขตพื้นที่ชั้นในของกรุงเทพมหานครเริ่มตั้งแต่ สะพานซังฮี้ ถนนราชวิถี อนุสาวรีย์ชัยสมรภูมิ แยกพระรามที่ 9 ถนนรัชดาภิเษก แยกคลองเตย ถนนพระรามที่ 4 แยกวิทยุ ถนนสาทร และจรดสะพานสมเด็จพระเจ้าตากสิน รวมทั้งพื้นที่ครอบคลุมสัญญาณไฟจราจร จำนวน 143 ทางแยกและ 3 ทางข้าม

ระยะที่ 2 ครอบคลุมพื้นที่ 150 ตารางกิโลเมตรในวงแหวนชั้นกลางของกรุงเทพมหานครเริ่มตั้งแต่ สะพานกรุงเทพ ถนนรัชดาภิเษก ถนนจรัลสนิทวงศ์ ถนนวงศ์สว่างถนนประชาชื่น ถนนแจ้งวัฒนะ ถนนรามอินทรา ถนนสุขาภิบาล1 ถนนศรีนครินทร์ถนนสุขุมวิท ถนนพระรามที่ 3 และจรดสะพานกรุงเทพ รวมทั้งพื้นที่ครอบคลุมสัญญาณไฟจราจรจำนวน 226 ทางแยก รวมกับระยะที่ 1 เป็น 369 ทางแยก

ระยะที่ 3 ครอบคลุมทั่วทั้งกรุงเทพมหานครมีทางแยกรวม 600 ทางแยก

หมายเหตุ ข้อมูลทั้งหมดอ้างอิงจาก Website http://www.ocmlt.go.th, http://www.peek-traffic.com และ http://www.trafficlinq.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข-3 แสดงระยะของระบบ ATC ที่ใช้ในกรุงเทพมหานคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Aaron M. Tenenbaum and Moshe J. Augenstein (1981) : *"Data Strcuture Using Pascal"*, Prentice-Hall Software Series, 2 1986

[2] George F. Luger and Willium A. Stubblefield : *"Artificial Intelligence"*, Addidon Wesley Longman Inc., 3 1998

[3] อาจารย์อภิเนตร อูนากูล : *"Object-Oriented Analysis and Design"*, แผนกตำรา คณะวิศวะกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 1 2000

[4] ดร. วีระศักดิ์ ซึงถาวร : *"Fundamental of JAVA Programming volume 1"*, Sum publishing, 1998

[5] ดร. วีระศักดิ์ ซึงถาวร : *"Fundamental of JAVA Programming volume 2"*, ซีเอ็ดยูเคชั่น, 2000

[6] กิตติ ภักดีวัฒนะกุล : *"JAVA ฉบับโปรแกรมเมอร์"*, KTP comp & consult, 3 2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้