ระบบควบคุมและดูแลการเดินรถจำลอง

TRAIN OPERATION SYSTEM SIMULATOR

คณิน คงชาตรี
KANIN KONGCHATREE
บุญญาฤทธิ์ ทรัพย์เจริญ
BOONYARIT SUBCHAROEN
กรวิชย์ สกุลพุทธิไพบูลย์
KORAWIT SAKULPUTTIPAIBUL

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ระบบควบคุมและดูแลการเดินรถจำลอง

TRAIN OPERATION SYSTEM SIMULATOR

คณิน คงชาตรี

KANIN KONGCHATREE

บุญญาฤทธิ์ ทรัพย์เจริญ

BOONYARIT SUBCHAROEN

กรวิชย์ สกุลพุทธิไพบูลย์

KORAWIT SAKULPUTTIPAIBUL

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRAIN OPERATION SYSTEM SIMULATOR

KANIN KONGCHATREE

BOONYARIT SUBCHAROEN

KORAWIT SAKULPUTTIPAIBUL

THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING
FACULTYN OF ENGINEERING
KINGS MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	ระบบควบคุมและดูแลการเดินรถจำลอง
Thesis Title	TRAIN OPERATION SYSTEM SIMULATOR
ชื่อนักศึกษา	นายคณิน คงชาตรี รหัสนักศึกษา 56010116
	นายบุญญาฤทธิ์ ทรัพย์เจริญ รหัสนักศึกษา 56010694
	นายกรวิชย์ สกุลพุทธิไพบูลย์ รหัสนักศึกษา 56011492
ระดับปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
ปริญญานิพนธ์ปีการศึกษา	2559

(.....)
ผศ. มยุรี เลิศเวชกุล
อาจารย์ที่ปรึกษาปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบจำลองการควบคุมและดูแลการเดินรถ		
Thesis Title	Train Operation System Simulator		
ชื่อนักศึกษา	นายคณิน คงชาตรี	รหัสนักศึกษา	56010116
	นายกรวิชย์ สกุลพุทธิไพบูลย์	รหัสนักศึกษา	56011492
	นายบุญญาฤทธิ์ ทรัพย์เจริญ	รหัสนักศึกษา	56010694
ระดับปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมสารสนเทศ		
ภาควิชา	วิศวกรรมคอมพิวเตอร์		
ปีการศึกษา	2559		
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.มยุรี เลิศเวชกุล		

บทคัดย่อ

ระบบจำลองการควบคุมและดูแลการเดินรถ (Train Operation System Simulator) เป็นการจำลองสภาพแวดล้อมให้ผู้อบรมได้ฝึกฝนและเรียนรู้ขั้นตอนการปฏิบัติอย่างถูกต้อง เพื่อให้เกิดความชำนาญในการปฏิบัติงานและสามารถแก้ปัญหาในสถานการณ์ต่าง ๆ ได้อย่างมีประสิทธิภาพ โดยยึดความปลอดภัยเป็นหลัก โดยได้มีการศึกษาถึงลำดับขั้นตอนระเบียบวิธีจนถึงลำดับในการปฏิบัติเมื่อเกิดเหตุการณ์ผิดปกติขึ้น

โดยระบบจำลองนี้จะแสดงเส้นทางการเดินรถตามตารางการเดินรถ และควบคุมให้มีขั้นตอนการปฏิบัติงานในการควบคุมการเดินรถ ตั้งแต่การจองเส้นทางในรูปแบบของ track ต่าง ๆ การแบ่งช่วงของราง (section) การเดินทางเข้าออกในแต่ละสถานี รวมถึงระบบอาณัติสัญญาณต่าง ๆ ที่ใช้ในการเดินรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Train Operation System Simulator		
Student	Mr. Kanin Kongchatree	Student ID.	56010116
	Mr. Korawit Sakulputtipaibul	Student ID.	56011492
	Mr. Boonyarit Subcharoen	Student ID.	56010694
Degree	Bachelor of Engineering		
Program	Information Engineering		
Department	Computer Engineering		
Academic Year	2016		
Thesis Advisor	Asst.Prof. Mayuree Lertwatechakul		

ABSTRACT

This project is about developing a simulator for Train Operation System. the simulation features allow an operator to learn how to operate system properly and to solve problems in different situations followed by the protocol.

The software can to evaluate a train operator in term of decision making , working procedure and how long they take to solve a problem. As to have a good mindset of knowledge and decision making for an operator.

By this software will show the traveling route that user select from traveling timetable and control to have procedures in train operation control. Such as track reserve , track segmentation , getting in and out in each station and also train signaling.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จได้ด้วยดีเป็นผลเนื่องมาจากความร่วมมือของคณะผู้จัดทำและความกรุณาจากอาจารย์ที่ปรึกษา ผศ.มยุรี เลิศเวชกุล ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และคำแนะนำที่ดีแก่ผู้จัดทำ ทางผู้จัดทำขอขอบพระคุณ

สุดท้ายนี้คณะผู้จัดทำขอขอบคุณ บิดา มารดา ผู้มีพระคุณทุกท่านที่ไม่ได้กล่าวถึงในที่นี้ ที่ให้การสนับสนุนด้านต่างๆ และเพื่อนๆ พี่ๆ น้องๆ ที่ได้ให้กำลังใจและให้การสนับสนุนในการทำปริญญานิพนธ์ครั้งนี้มาโดยตลอด

คณิน คงชาตรี

กรวิชย์ สกุลพุทธิไพบูลย์

บุญญาฤทธิ์ ทรัพย์เจริญ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

โครงงานระบบจำลองการควบคุมและดูแลการเดินรถถูกคิดค้นขึ้นมาเพื่อฝึกสอนและให้ความรู้เรื่องระบบอาณัติสัญญาณ (Signaling System) ให้ผู้เข้าอบรมหรือผู้ใช้งานได้เรียนรู้ระบบจำลองสภาวะแวดล้อมการทำงานของระบบรถไฟด้วยโปรแกรมจำลองกิจกรรมภายในห้องควบคุมกลาง (Operations Control Center) เพื่อให้ผู้เข้าอบรมในการควบคุมรถไฟได้ลองสั่งการไปยังรถไฟและอุปกรณ์ที่อยู่ภายใต้ขอบเขตการทำงานของตนเองแล้วรถไฟและอุปกรณ์ต่างๆในโปรแกรมระบบจำลองจะทำงานไปตามหลักการของระบบที่ใกล้เคียงกับสภาพความเป็นจริง ทำให้ผู้ใช้สามารถทำความเข้าใจกับขั้นตอนการดำเนินการทั้งในสภาวะปกติและไม่ปกติจากข้อมูลสภาวะแวดล้อมจำลองที่สร้างขึ้น เช่น ข้อมูลเฉพาะของขบวนรถ ข้อมูลของราง ตำแหน่งและสถานะของอุปกรณ์อาณัติสัญญาณ ข้อมูลของสภาพภูมิประเทศ เป็นต้น เพื่อให้เกิดความคุ้นเคยและความชำนานในการตัดสินใจเพื่อที่จะสามารถนำไปใช้งานจริงได้

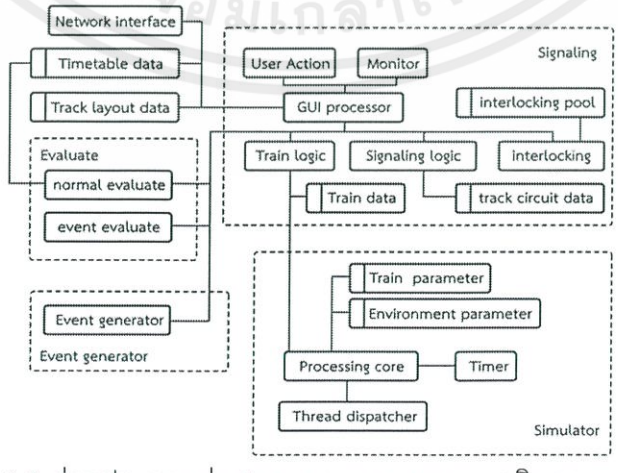
1.2 วัตถุประสงค์ของโครงงาน

โครงงานนี้จัดทำขึ้นเพื่อให้ผู้ใช้โปรแกรมจำลองได้ฝึกใช้ศึกษาและเรียนรู้การทำงานของรถไฟและระบบอาณัติสัญญาณต่างๆให้เกิดความชำนาญ สามารถนำไปเป็นโปรแกรมต้นแบบเพื่อพัฒนาต่อตลอดจนถึงสามารถนำไปใช้ได้จริงในการฝึกอบรม

1.3 ภาพรวมหรือโครงสร้างรวมของโครงงาน

โครงงานนี้มีองค์ประกอบทางด้าน ซอฟต์แวร์(Software)
แบ่งออกเป็น 4 module หลักๆคือ

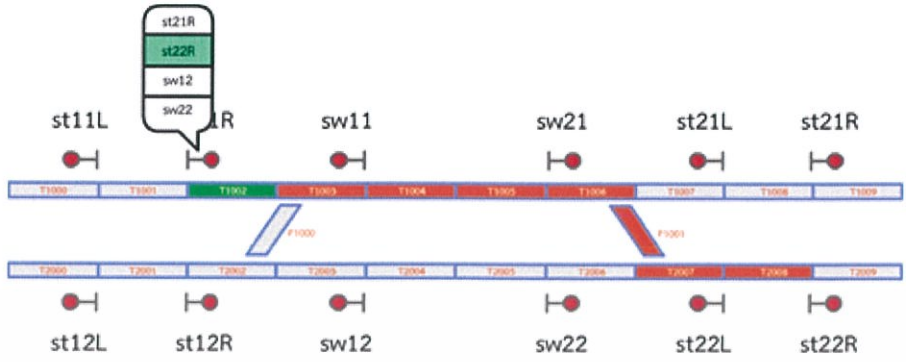
1.) Train Module โมดูลหลักสำหรับจำลองการทำงานของระบบรถไฟ คือส่วนแกนหลักที่ใช้ในการจำลองข้อมูลเหตุการณ์ต่างๆ ของระบบควบคุมการเดินรถ



รูปที่ 1.1 ส่วนประกอบต่างๆ ของระบบควบคุมการเดินรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) Signaling Module โมดูลจำลองการทำงานของระบบอาณัติสัญญาณ



รูปที่ 1.2 การทำงานของการเลือกเส้นทางและสถานะขององค์ประกอบต่างๆ ในระบบ

3.) Event Generator Module ตัวสร้างเหตุการณ์ คือส่วนที่จำลองข้อมูลสถานะขององค์ประกอบต่างๆ ในระบบการเดินรถไฟ ที่ทำให้เกิดสถานการณ์ที่ผิดปกติในระบบ เพื่อเป็นโจทย์ให้ผู้เข้าอบรมต้องคิดและลงมือแก้ไขสถานการณ์

4.) Evaluate Module เครื่องมือประเมิน เป็นส่วนระบบประเมินประสิทธิภาพการทำงานในด้านการควบคุมการเดินรถไฟของผู้เข้าอบรม เพื่อตรวจวัดประสิทธิภาพในการทำหน้าที่เป็นผู้ควบคุมการเดินรถของผู้เข้าอบรมแต่ละครั้งว่ามีประสิทธิภาพเพียงใดและสามารถแก้ไขสถานการณ์แต่ละประเภทได้หรือไม่และวิธีการที่เลือกใช้เป็นวิธีการที่ถูกต้องหรือมีประสิทธิภาพเหมาะสมหรือไม่ โดยระบบจำลองจะแสดงผลการใช้งานของผู้อบรมบนหน้าต่างของโปรแกรมระบบจำลองการเดินรถเป็นเวลาที่ผู้ทดลองใช้ควบคุมรถในแต่ละสถานี

5.) Track Reservation module เป็นอีก module ที่ช่วยให้ผู้ใช่สามารถใช้งานโปรแกรมได้ง่ายขึ้น เนื่องจากทำให้โปรแกรมสามารถเปลี่ยนสถานะของรางต่างๆได้หลายๆรางในครั้งเดียวไม่ต้องเปลี่ยนเองทีละรางและยังสามารถเคลียร์ค่าสถานะของรางต่างๆได้ในครั้งเดียวเช่นกันเพื่อแก้ไขเมื่อมีการใช้งานการจองรางผิดหรือเมื่อต้องการแก้ไขก็ช่วยให้แก้ได้ง่ายขึ้น

6.) Track module คือระบบที่ใช้สร้างเส้นทางการเดินรถให้กับ Train module ที่มีอยู่ว่ารถจะวิ่งตามเส้นทางไหนบ้างหรือการเดินรถไปยังสถานีต่างๆแล้วรถขบวนไหนต้องวิ่งบนเส้นทางใดบ้าง ภายในโปรแกรมจะมีการเก็บค่าสถานะต่างๆของรางไว้ สามารถเปลี่ยนแปลงได้ให้จองไว้สำหรับรถขบวนต่างๆได้ และยังนำไปใช้กับ Even Generator Module อีกด้วย (สร้างเหตุการณ์ต่างๆบนรางรถไฟที่รถจะต้องวิ่งผ่านเพื่อให้ผู้ใช้ทำการแก้ปัญหา)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินงานโครงงาน

Project I

-การดำเนินงานในช่วงนี้จะเริ่มดำเนินการในส่วนของการศึกษาข้อมูลที่จำเป็นต้องนำมาประยุกต์ใช้ในโครงงาน เช่นรูปแบบตารางการเดินของรถไฟ device และ hardware ต่างๆ ที่เกี่ยวข้องกับระบบบอาณัติสัญญาณ คำสั่งต่างๆ เหตุการณ์ที่อาจเกิดขึ้นได้ระหว่างการทำงานของรถไฟ แล้วจึงนำมาวางแผนการเขียนโปรแกรมจำลองให้สามารถใช้งานได้ในระดับหนึ่ง เช่นสามารถสั่งรถไฟให้เดินทางจากจุดหนึ่งไปยังจุดหนึ่งในโปรแกรมจำลองได้ (simulator module และ Signaling Module)

Project II

-หลังจากที่ได้ออกแบบและเขียนโปรแกรมจำลองขึ้นมาแล้วในส่วนของ project I เราก็จะมาเพิ่มส่วนของโปรแกรมที่จะทำการสร้างเหตุการณ์เสมือนขึ้นมาเพื่อให้ผู้ใช้หรือผู้เข้าอบรมการใช้งานได้ฝึกการทำงานในสถานการณ์ที่อาจเกิดขึ้นและโปรแกรมคำนวณระดับการทำงานของผู้ใช้ว่ามีประสิทธิภาพอย่างไรในการทำงานต่อเหตุการณ์ต่างๆที่โปรแกรมได้จำลองเสมือนขึ้นมา(Event Generator Module, Evaluate Module)

แผนผัง หรือตารางเวลาการดำเนินงานโครงงาน

ตารารงที่ 1.1 ตารางแสดงแผนการดำเนินงานตามขอ 4 (สิงหาคม 2559 – เมษายน 2559)

ขั้นตอนการดำเนินงาน	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม	มกราคม	กุมภาพันธ	มีนาคม	เมษายน
ศึกษาข้อมูลเบื้องต้นและวางแผนการเขียน simulator และsignaling module	←	—	—	→					
เขียนโปรแกรม simulator			←	—	—	—	→		

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

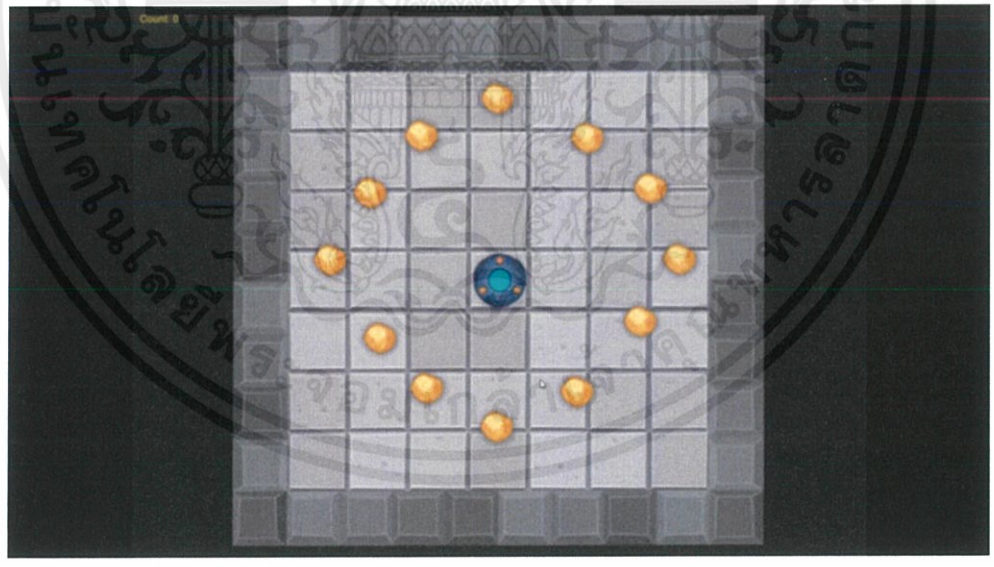
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีพื้นฐานที่ต้องใช้

2.1 Unity game engine

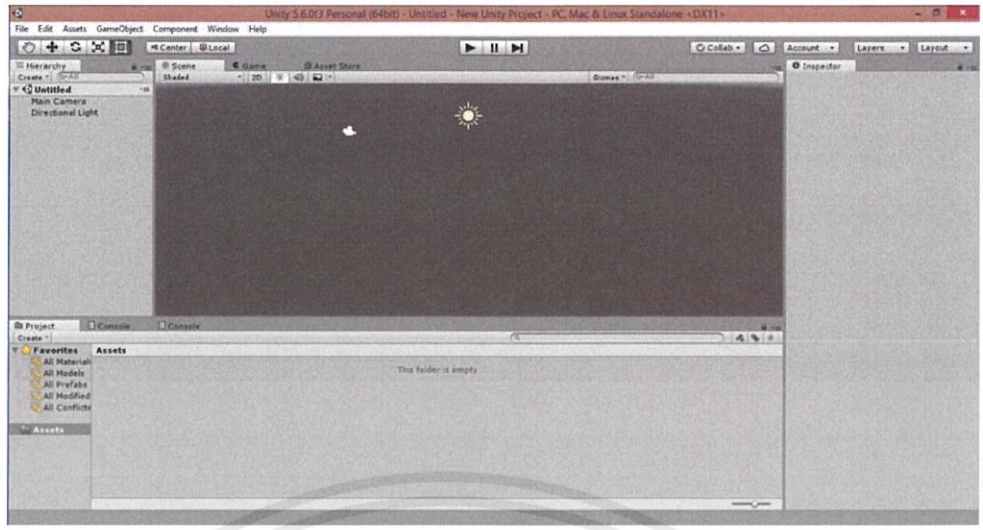
2.1.1 Unity คืออะไร

Unity คือเกมเอนจิ้นสำหรับการสร้างเกม ที่ถูกพัฒนาขึ้นมาอย่างต่อเนื่อง ตั้งแต่ในช่วงแรกๆ ที่รองรับการพอร์ทเกมลงบน Windows, OS X และเว็บเท่านั้น แต่ปัจจุบันได้เพิ่มความสามารถในการพอร์ทลงบน iOS,Androidและแพลตฟอร์มอื่นๆเกือบทุกแพลตฟอร์ม ปัจจุบันโปรแกรม Unity สามารถทำงานได้ทั้งบน Windows และ OS X จุดเด่นของ Unity นั้นโดดเด่นกว่าเกมเอนจิ้นตัวอื่นๆเป็นอย่างมาก เพราะนอกจากความง่ายในการใช้งาน ความสามารถในการพอร์ทลงบนแพลตฟอร์มต่างๆ คุณภาพของเกมที่ได้ก็อยู่ในระดับสูงอีกด้วยนอกจากการมีเวอร์ชันฟรีให้ใช้งานแล้ว ราคาค่า license ของ Unity เองก็ถือว่าถูกมาก หากเทียบกับเกมเอนจิ้นตัวอื่นดังนั้นจึงไม่แปลกเลยที่ปัจจุบัน Unity คือเกมเอนจิ้นอันดับหนึ่ง และมีผู้ใช้งานมากที่สุด เกมที่อยู่ใน App Store และ Google Play เกือบครึ่งถูกสร้างด้วย Unity และยังมี plugin สำเร็จรูปให้เลือกใช้มากมายอย่างครบถ้วนไม่ว่าจะเป็น การเคลื่อนไหวของวัตถุ หรือ ความเร็ว ความเร่ง function ใน ฟิสิกส์ ต่างๆ และอื่นๆ อีกมากมายบนพื้นฐานของภาษา C#



รูปที่ 2.1 ตัวอย่างเกมที่เขียนด้วย unity game engine

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ตัวอย่างโปรแกรม unity game engine

2.1.2 ตัวอย่างฟีเจอร์หลักๆ ของ unity

2.1.2.1 Rigid body

เมื่อใส่ component rigid body เข้าไปในออปเจ็ค นั้นๆ แล้วออปเจ็คนั้นก็จะมีแรงโน้มถ่วงเกิดขึ้น เมื่อ Run ก็จะเห็นว่า ออปเจ็ค นั้นมันร่วงลงไปเรื่อย ซึ่งเป็นเพราะมันมีน้ำหนักขึ้นมาแล้วนั้นเอง ซึ่งวิธีการนั้น ให้เลือกออปเจ็ค ที่จะทำการใส่ rigid body จากนั้นไปที่แถบเมนู แล้วเลือก Component แล้วก็เลือก Physics แล้วเลือก Rigid body

2.1.2.2 Run sprite

เป็นฟังก์ชั่นที่ทำให้ภาพตัวสไปรต์เคลื่อนไหวได้เหมือนเป็นแอนนิเมชันใช้กับพวกตัวละครต่างๆ ในเกมเช่นตัวละครที่ผู้เล่นต้องทำการบังคับแล้วมีการเปลี่ยนแปลงหน้าตาเมื่อเกินเหตุการณ์ต่างๆ

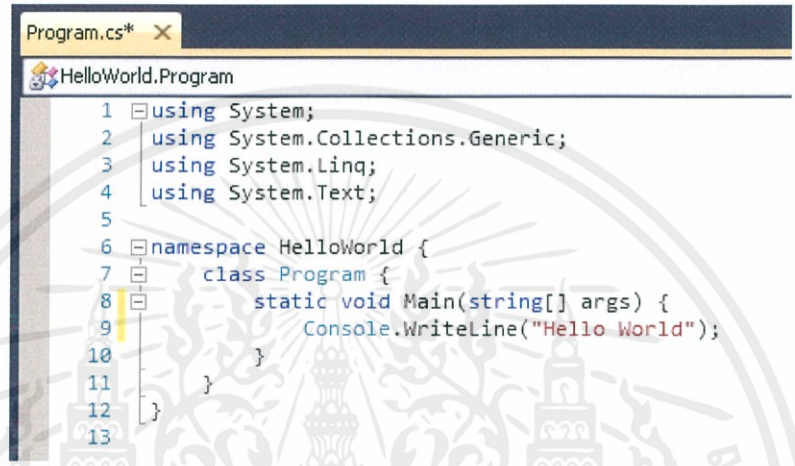
2.2 ภาษา C#

2.2.1 C# คืออะไร

C# คือ ภาษาคอมพิวเตอร์ประเภท object-oriented programming พัฒนาโดย Microsoft โดยมีเป้าหมายเพื่อใช้ในการรวมความสามารถการคำนวณของ C++ ด้วยการโปรแกรมง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ Java C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคือ อำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บและทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึงออปเจ็ค ของโปรแกรมหรือเมธอด โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปราศจากความต้องการให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอกซ้ำ ภาษา C# ถูกพัฒนาขึ้นมาภายใต้พื้นฐานของ .NET Framework เป็นการการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi , ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP (โปรแกรมเชิงวัตถุ) มากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลงและมีสิ่งที่เกินความจำเป็นน้อยลง C# ถูกรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO และปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง



รูปที่ 2.3 ตัวอย่างคำสั่งภาษา C#

2.2.2 เปรียบเทียบภาษา C# กับภาษาอื่นๆ

เมื่อพูดถึงความใกล้เคียงกับภาษาอื่นๆ ภาษา C# ใกล้เคียงกับภาษา Java มากที่สุด โดยมีความเหมือนกันถึง 70% ดั้งนั้นนักเขียนโปรแกรมภาษา Java จึงอาจย้ายมาเขียนภาษา C# ได้โดยศึกษาว่ามีสิ่งใดที่แตกต่างกันบ้าง ภาษา C# ยังมีความคล้ายคลึงกับภาษ C++.NET และภาษา VB.NET เป็นอย่างมาก ทำให้นักเขียนโปรแกรมภาษา C# สามารถอ่าน-เขียนโค้ดในภาษากลุ่มนี้ได้เมื่อฝึกฝนเพียงเล็กน้อย

นอกจากนั้น C# และภาษา Java ทั้งคู่เป็นแบบสืบจากคลาสหลักได้คลาสเดียว ขณะที่ภาษา C++ สามารถสืบจากคลาสหลักได้มากกว่าหนึ่ง (Multiple inheritance) โดยภาษา C# และภาษา Java ใช้ Interface มาทดแทน Multiple inheritance เหมือนกันทั้งคู่

และสิ่งที่ภาษา C# และ Java มีร่วมกันคือเรื่อง Garbage Collection แต่ไม่มีใน C++ จึงทำให้ดูเหมือนว่าภาษา Java ต่อยอดมาจากภาษา C++ และ C# ต่อยอดมาจาก Java อีกที ที่เป็นเช่นนั้นเพราะทั้ง Java และ C# มีต้นสายมาจาก C++ ทำให้สองภาษานี้ดูคล้ายกัน แต่ภาษา C# ไม่ใช่ภาษา Java มันมีกลไกที่เป็นเอกลักษณ์หลายอย่าง เช่น พารามิเตอร์แบบ reference และ output การจัดเก็บออปเจ็ค ไว้ใน stack การทำ Versioning และยังมีสิ่งใหม่ๆ ที่เป็นข้อดี เช่น delegate, properties และ operator overloading ซึ่งจะไม่พบในภาษา Java

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ข้อดีของภาษา C#

- Component oriented - เป็นภาษาที่เน้นชิ้นส่วนโดยถูกออกแบบมาเป็นอย่างดีทำให้สามารถนำมาใช้ต่อกันเป็นอะไรก็ได้
- สิ่งต่าง ๆ ใน C# เป็นออบเจ็กต์ทั้ง Component oriented เป็นภาษาที่เน้นชิ้นส่วนโดยถูกออกแบบมาเป็นอย่างดีทำให้สามารถนำมาใช้ต่อกันเป็นอะไรก็ได้ นอกจากนั้นสิ่งต่าง ๆ ใน C# เป็นออบเจ็กต์ทั้งหมด
- เป็นภาษา ที่ทนทาน (robust) ทนต่อความผิดพลาด ไม่ทำให้ระบบเสียหายหรือระบบทำงานช้า เพราะ C# มีข้อดีคือ garbage collection , exception , type-safety และ versioning
- ภาษา C# จัดเตรียมกลไกไว้หลายอย่างที่ช่วยให้ผู้เขียนโปรแกรมสามารถนำโค้ดที่เขียนไว้ใน project หนึ่งไปใช้กับอีก project หนึ่งได้ง่าย นอกจากนั้นภาษา C# ยังสามารถเรียกใช้คลาสหลายพันคลาสใน .NET Framework ได้โดยตรง ทำให้ลดเวลาการพัฒนาซอฟท์แวร์ได้มากหมด
- เป็นภาษา ที่ทนทาน (robust) - ทนต่อความผิดพลาด ไม่ทำให้ระบบขัดข้องหรือระบบทำงานช้า เพราะ C# มีข้อดีคือ garbage collection , exception , type-safety และ versioning
- ภาษา C# จัดเตรียมกลไกไว้หลายอย่างที่ช่วยให้ผู้เขียนโปรแกรมสามารถนำโค้ดที่เขียนไว้ในโปรเจค หนึ่งไปใช้กับอีกprojectหนึ่งได้ง่าย นอกจากนั้นภาษา C# ยังสามารถเรียกใช้คลาสหลายพันคลาสใน .NET Framework ได้โดยตรง ทำให้ลดเวลาการพัฒนาซอฟท์แวร์ได้มาก

2.2.4 โครงสร้างของภาษา C#

2.2.4.1 ตัวอย่างโปรแกรมในภาษา C#

โครงสร้างพื้นฐานของการเขียนโปรแกรมภาษา C# จะสร้างโปรแกรมที่ชื่อว่า Hello World ซึ่งจะแสดงคำว่า "Hello Word" ออกทางหน้าจอคอมพิวเตอร์ ตัวโปรแกรมจะมีชุดคำสั่งดังนี้

```
// #C Hello Word program
using System;

namespace Hello
{
   class Program
   {
      static void Main()
      {
         // print text to the screen
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Console.WriteLine("Hello World!");
        }
    }
}
```

ผลลัพธ์ของโปรแกรมจะได้ออกมาเป็นคำว่า Hello World!

2.2.4.2 คำสั่ง using

คำสั่ง using ถูกใช้เพื่อ include ไลบรารี่และฟังก์ชันในภาษา C# ทั้งฟังก์ชันมาตรฐานและฟังก์ชันที่ผู้ใช้สร้างขึ้นเอง จากตัวอย่างโค้ด Using System หมายถึงได้ทำการ include คลาสและฟังก์ชันทั้งหมดภายใต้ System namespace เพื่อที่จะนำมาใช้ในโปรแกรม

2.2.4.3 ฟังก์ชัน Main

ภาษา C# นั้นต้องการฟังก์ชันที่เรียกว่า Main() เพื่อเริ่มต้นและสิ้นสุดโปรแกรม คุณสามารถเขียนโค้ดโปรแกรมของคุณในฟังก์ชัน Main และใช้เรียกฟังก์ชันอื่นๆ และฟังก์ชัน Main นั้นจะมีหลายแบบ ซึ่งคุณสามารถสร้างได้โดยใช้คำสั่ง void, int และอื่นๆ โดยที่คำสั่ง void บ่งบอกว่าฟังก์ชันนี้ไม่ได้มีค่าส่งหลับ น่าไม่เช่นนั้น คุณต้องใช้คำสั่ง return เพื่อส่งค่ากลับหลังจากจบฟังก์ชัน อย่างไรก็ตามคุณจะได้เรียนรู้เกี่ยวกับฟังก์ชันในภาษา C# ภายหลัง

2.2.4.4 คำสั่ง namespace

คำสั่ง namespace นั้นถูกใช้สำหรับการประกาส namespace สำหรับคลาส ในภาษา C# namespace สามารถประกอบไปด้วยคลาสเดียวหรือหลายคลาสก็ได้ ในตัวอย่างที่ได้สร้าง namespace Hello สำหรับโปรแกรม

2.2.4.5 การสร้างคลาส

คลาสเป็นเหมือนโค้ดที่สามารถยืดขยายหรือปรับแต่งได้ คลาสเป็นเหมือนโค้ดที่สามารถยืดขยายหรือปรับแต่งได้ มันเป็นเทมเพลตสำหรับเอาไว้สร้างออบเจ็ค หรือกำหนดค่าเริ่มต้น ซึ่งมันจะมีแอทริบิวต์และเมธอดที่เป็นสมาชิกของคลาส ยกตัวอย่างเช่น คำสั่ง class Program ให้เพื่อสร้างคลาสที่มีชื่อว่า Program

2.2.4.6 Comment

ในภาษา C# คุณสามารถคอมเม้นต์โดยการใช้เครื่องหมาย// ก่อนหน้าบรรทัดของตัวอักษรที่ต้องการคอมเม้นต์ ผลลัพธ์ของการคอมเมนต์ก็คือโค้ดนั้นจะไม่มีผลกับโปรแกรม // และจะถูกเพิกเฉยจากคอมไพเลอร์ และการคอมเม้นต์แบบนี้เรียกว่าการคอมเม้นต์แบบ 1 บรรทัด มาดูตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// This is my first comment
```

นอกจากนี้คอมเม้นท์อีกแบบหนึ่งคือ การคอมเม้นท์แบบบล็อคหรือหลายบรรทัด ซึ่งจะทำให้คุณสามารถคอมเม้นท์โค้ดได้มากกว่า 1 บรรทัดหรือค้อมเม้นท์ในส่วนที่ซับซ้อนขึ้น โดยสิ่งที่คุณคอมเม้นท์จะอยู่ในเครื่องหมาย /* และ */ และนี่เป็นตัวอย่างการคอมเม้นท์แบบหลายบรรทัด

2.2.4.7 Keywords

keyword เป็นคำสงวนที่คุณไม่สามารถใช้ได้ในส่วนของโปรแกรมสำหรับการกระทำบางอย่าง เช่น การใช้ในการประกาศชื่อตัวแปรและเมธอด ในภาษา C# มีคำสงวนมากมายซึ่งแสดงในรายการข้างล่างนี้

รายการของ keyword ในภาษา C#

- abstract
- as
- base
- bool
- break
- byte
- case
- catch
- char
- checked
- class
- const
- continue
- decimal
- default delegate
- do
- double else
- enum
- event
- explicit extern
- false
- finally
- fixed
- float

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- for
- foreach
- goto
- if
- implicit
- in
- in (generic modifier)
- int
- interface
- internal
- is
- lock
- long
- namespace
- new
- null
- object
- operator
- out
- out (generic modifier)
- override
- params
- private
- protected
- public
- readonly
- ref
- return
- sbyte
- sealed
- short
- sizeof
- stackallocstatic
- string

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- struct
- switch
- this
- throw
- true
- try
- typeof
- uint
- ulong
- unchecked
- unsafe
- ushort
- using
- virtual
- void
- volatile
- while

2.2.5 ตัวแปรและประเภทข้อมูล

2.2.5.1 ตัวแปร

ตัวแปรถูกใช้เพื่อเก็บข้อมูลในหน่วยความจำเพื่อนำข้อมูลเหล่านั้นไว้ใช้ภายหลังในโปรแกรม ในภาษา C# นั้นมีตัวแปรประเภทต่างๆ ที่มีชนิดข้อมูลที่แตกต่างกันไป เช่น Boolean integer floating point string และพอยน์เตอร์ โดยการประกาศตัวแปรนั้นมีรูปแบบดังนี้ type identifier; (ด้านหน้าเป็นชนิดของตัวแปลที่ต้องการให้กำหนดไว้ด้านหน้า ด้านหลังคือชื่อของตัวแปลที่ต้องการตั้ง)

โดยที่ type เป็นชนิดของข้อมูลพื้นฐานที่มีในภาษา C# แต่อย่างไรก็ตามมันยังสามารถเป็นชนิดข้อมูลแบบอื่นๆ ได้ เช่น ออปเจ็คซึ่งจะกล่าวภายหลังในบทเรียนนี้ ส่วน identifier เป็นชื่อของตัวแปรที่ต้องการสร้างขึ้น

ในการตั้งชื่อตัวแปรนั้น ชื่อของตัวแปรไม่สามารถตรงกับคำสงวนที่มีในภาษา C# ได้ และจะเป็นแบบ case-sensitive นั่นหมายความว่า ตัวแปร name และ NAME เป็นตัวแปรคนละตัวแปรกัน ในการตั้งชื่อตัวแปรในภาษา C# จะมีกฎดังนี้

ชื่อของตัวแปรสามารถประกอบไปด้วย ตัวอักษรภาษาอังกฤษ ตัวเลข และเครื่องหมาย _ เท่านั้น และไม่สามารถขึ้นต้นด้วยตัวเลขได้และชื่อของตัวแปรต้องไม่ตรงกับคำสงวนในภาษา C# แต่อย่างไรก็ตาม คุณสามารถใช้เครื่องหมาย @ นำหน้าชื่อตัวแปรได้หากคุณต้องการใช้คำสงวน แต่นั่นไม่ใช่วิธีการปฏิบัติที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกฎการตั้งชื่อของตัวแปรที่ได้กล่าวมานั้นยังสามารถใช้กับการตั้งชื่อ เมธอด คลาส Interfaces หรือสิ่งต่างๆ ที่ผู้ใช้เป็นคนกำหนดขึ้นมา ซึ่งคุณจะได้เห็นตัวอย่างการใช้งานในบทต่อๆ ไปของบทเรียน

2.2.5.2 ประเภทข้อมูล

ในภาษา C# มีประเภทข้อมูลเพียงพอที่ให้สามารถจัดการกับข้อมูลประเภทต่างๆ ได้ เช่น ตัวอักษร ข้อความ ตัวเลขจำนวนเต็ม และจำนวนจริง เป็นต้น ซึ่งข้อมูลแต่ละประเภทจะใช้สำหรับเก็บค่าที่แตกต่างกันออกไป เช่น เก็บค่าคะแนนของผู้เล่นเกมไว้ในตัวแปรประเภทจำนวนเต็ม หรือเก็บชื่อไว้ในตัวแปรประเภทข้อความ เป็นต้น

และนี่เป็นข้อมูลพื้นฐาน 4 ประเภทที่มีในภาษา C# ซึ่งประเภทข้อมูลเหล่านี้เป็น Primitive data type หรือประเภทข้อมูลพื้นฐานในการเขียนโปรแกรม

- Characters: นี่เป็นประเภทของข้อมูลที่ใช้ในการเก็บตัวอักษร โดยการใช้คำสั่ง char หรือ string ในการประกาศตัวแปร
- Integer: นี่เป็นประเภทข้อมูลที่ใช้ในการเก็บข้อมูลตัวเลขแบบจำนวนเต็ม โดยการใช้คำสั่ง int หรือ long ในการประกาศตัวแปร แต่ที่แตกต่างกันคือหน่วยความจำที่ใช้ในการเก็บ เช่น long จะเก็บข้อมูลได้มากกว่า int และมันก็ใช้หน่วยความจำมากกว่าเช่นกัน
- Floating point: นี่เป็นประเภทของข้อมูลที่ใช้สำหรับการเก็บตัวเลขแบบทศนิยมหรือจำนวนจริง โดยการใช้คำสั่งอย่างเช่น float หรือ double ในการประกาศตัวแปร
- Boolean: ประเภทข้อมูลนี้สามารถเก็บข้อมูลได้เพียงแค่สองค่าคือ true และ false

ประเภทของข้อมูลแบบอื่นในภาษา C# นั้นจะเป็นประเภทข้อมูลแบบออบเจ็คอาเรย์ ซึ่งจะมากับไลบรารี่ของภาษาหรือผู้ใช้ก็สามารถสร้างขึ้นเองได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส	ประเภท	คำอธิบาย	ค่า
Char	char	ตัวอักษร Unicode character ขนาด 16 bit	U +0000 ถึง U +ffff
SByte	sbyte	เลขจำนวนเต็มขนาด 8 bit	-128 ถึง 127
Int16	short	เลขจำนวนเต็มขนาด 16 bit	-32,768 ถึง 32,767
Int32	int	เลขจำนวนเต็มขนาด 32 bit	-2,147,483,648 ถึง 2,147,483,647
Int64	long	เลขจำนวนเต็มขนาด 64 bit	-9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807
Single	float	เลขจำนวนจริงขนาด 32 bit	-3.40282E38 ถึง 3.40282E38
Double	double	เลขจำนวนจริงขนาด 64 bit	-1.79769E308 ถึง 1.79769E308
Byte	byte	เลขจำนวนเต็มบวกขนาด 8 bit	0 ถึง 255
UInt16	ushort	เลขจำนวนเต็มบวกขนาด 16 bit	0 ถึง 65,535
UInt32	unsigned int	เลขจำนวนเต็มบวกขนาด 32 bit	0 ถึง 4,294,967,295
UInt64	unsigned long	เลขจำนวนเต็มบวกขนาด 64 bit	0 ถึง 18,446,744,073,709,551,615
Decimal	decimal	เลขขนาด 128 bit	-7.92282E28 ถึง 7.92282E28
Boolean	boolean	ค่า true หรือ false	true, false
String	string	ใช้เก็บตัวอักษรหลายตัว หรือ text	Multiple characters (Char array)

รูปที่ 2.4 ตารางแสดงประเภทของข้อมูลพื้นฐานในภาษา C#

2.2.5.3 การประกาศตัวแปร

ตามที่ได้อธิบายแนวคิดและวิธีการประกาศตัวแปรไปแล้ว ต่อไปมาดูตัวอย่างการประกาศและใช้งานตัวแปรในภาษา C#

```
int x;
x = 10;
```

ในตัวอย่าง ที่ได้ประกาศตัวแปรที่ชื่อว่า x ซึ่งเป็นตัวแปรแบบจำนวนเต็ม โดยใช้คำสั่ง int ในการประกาศ และบรรทัดต่อมาได้กำหนดค่าให้กับตัวแปร อย่างไรก็ตามสามารถประกาศตัวแปรและกำหนดค่าให้กับมันได้พร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int a = 4;
float b = 10.5;
String name = "Thomas";
```

ในตัวอย่างที่ได้ประกาศตัวแปรมา 3 ตัวแปร ตัวแปร a เป็นประตัวประเภท Integer และกำหนดค่า 4 ให้เป็นค่าของมัน ตัวแปร b เป็นตัวแปรประเภท float และกำหนดค่า 10.5 ให้กับมัน ตัวแปรตัวสุดท้าย name เป็นตัวแปรประเภท string และมีค่าคือ "Thomas"

2.2.5.4 Boolean

.Boolean เป็นประเภทข้อมูลที่สามารถเก็บได้สองค่าคือ true และ false มันใช้สำหรับเก็บข้อมูลที่มีค่าเป็นไปได้เพียงสองค่าหรือสองสถานะ เช่น เพศชายและหญิง กลางวันและกลางคืน เปิดหรือปิด เป็นต้น Boolean นั้นเป็นข้อมูลที่ใช้เป็น Expression ที่ทำงานร่วมกับคำสั่งควบคุมหรือคำสั่งเงื่อนไข

2.2.5.5 ประเภทข้อมูลตัวเลขจำนวนเต็ม

ในภาษา C# นั้น ข้อมูลประเภทจำนวนเต็มจะแบ่งเป็นหลายขนาด ซึ่ง Byte เป็นข้อมูลที่เก็บค่าได้น้อยที่สุด ส่วน Long จะเก็บค่าได้มากที่สุด และนอกจากนี้ยังมีประเภทแบบไม่มีเครื่องหมายด้วยที่เก็บเพียงจำนวนเต็มบวกเท่านั้นที่เรียกว่า unsigned

2.2.5.6 ประเภทข้อมูลตัวเลขทศนิยม

ประเภทข้อมูลแบบทศนิยมใช้สำหรับเก็บตัวเลขใดๆ ที่มีหลักของทศนิยมด้วย ข้อมูลประเภทนี้ใช้สำหรับเก็บข้อมูลที่มีความละเอียดหรือเศษส่วน เช่น ตัวเลขการทำควณทางวิทยาศาสตร์ ในภาษา C# ข้อมูลประเภทนี้จะมี float double และ decimal การกำหนดค่าให้กับตัวแปรยังสามารถใช้ในรูปแบบวิทยาศาสตร์ได้ ดังในคำสั่งข้างบนโดยการใช้ตัวอักษร E ซึ่งหมายถึงแทนสิบยกกำลัง ซึ่งในตัวอย่างหมายถึง 5.8786 x 10^12 หรือสามารถกำหนดค่าในรูปแบบของ Expression เหมือนในคำสั่ง double pi = 22 / 7.0f;

2.2.5.7 Char and String

ตัวแปรอีกประเภทหนึ่งที่สำคัญคือ Char และ String ซึ่งข้อมูลประเภทนี้มีรูปแบบการเก็บข้อมูลเหมือนกัน แต่สิ่งที่แตกต่างกันคือ Char จะเก็บข้อมูลเพียงหนึ่งตัวอักษร และใช้เครื่องหมาย ' ล้อมรอบค่าของ Literal ส่วน String สามารถเก็บได้หลายตัวอักษรและใช้เครื่องหมาย " ล้อมรอบค่าของ Literal

2.2.6 ค่าคงที่

ค่าคงที่ เป็นตัวแปรที่ค่าของมันไม่สามารถเปลี่ยนแปลงได้หลังจากที่ถูกสร้างขึ้น และค่าของมันจะต้องถูกกำหนดให้ในตอนที่สร้างทันที เนื่องจากมันคือตัวแปรประเภทหนึ่ง การใช้งานของมันจะเหมือนกับตัวแปรปกติ แต่ในการประกาศต้องใช้คำสั่ง const นำหน้าเพื่อบ่งบอกว่าเป็นตัวแปรค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประกาศค่าคงที่ในภาษา C# มีรูปแบบดังนี้

const type identifier = value;

const เป็นคำสั่งที่ใช้ในการประกาศค่าคงที่ type เป็นประเภทข้อมูลของค่าคงที่โดยมีหลักการเหมือนกับการประกาศตัวแปรทั่วไป identifier นั้นเป็นชื่อของค่าคงที่ โดยส่วนมากมักจะนิยมใช้เป็นตัวพิมพ์ใหญ่ทั้งหมด value เป็นค่าของค่าคงที่ที่สร้างขึ้นและต้องกำหนดพร้อมกับตอนประกาศค่าคงที่เสมอ

2.2.6.1 การประกาศค่าคงที่

เพื่อประกาศค่าคงที่ ให้ทำทุกอย่างเหมือนการประกาศตัวแปรปกติ แต่จะใช้คำสั่ง const คำหน้าและกำหนดค่าให้กับมันทันที

```
const int SIZE= 5;
const float PI = 3.14f;
const double G = 9.8;
```

ในตัวอย่าง ตามที่ได้สร้างค่าคงที่สามตัว ซึ่งมีประเภทข้อมูลแบบต่างๆ และได้กำหนดค่าให้กับมันในทันที โดยทั่วไปมักจะใช้อักษรตัวพิมพ์ใหญ่ในการตั้งชื่อให้กับค่าคงที่

2.2.7 ตัวดำเนินการ

ตัวดำเนินการ คือเครื่องหมายที่ใช้เพื่อให้ทำงานกับตัวแปรและค่าคงที่เพื่อสร้าง expression ในการเขียนโปรแกรม ในภาษา C#มีตัวดำเนินการประเภทต่างๆ ที่คุณจะต้องรู้จักก่อนที่จะเขียนโปรแกรม ข้างล่างนี้เป็นรายการของประเภทตัวดำเนินการที่คุณจะได้เรียนในบทเรียนนี้

- Assignment operator
- Arithmetic operators
- Compound assignment
- Increment and decrement
- Relational and comparison operators
- Logical operators
- Conditional ternary operator
- Bitwise operators
- Explicit type casting operator

2.2.7.1 Assignment operator

assignment operator หรือตัวดำเนินการกำหนดค่าใช้เครื่องหมายเท่ากับ = มันใช้สำหรับกำหนดค่าให้กับตัวแปรในภาษา C# ยกตัวอย่างเช่น:

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float weight = 58.3f;
int y = -3;
int x = y;
string name = "Marcus";
```

ในตัวอย่างด้านบนที่ได้ใช้ตัวดำเนินการกำหนดค่าเพื่อกำหนดค่าให้กับตัวแปรซึ่งมีประเภทข้อมูลแบบต่างๆ

2.2.7.2 Arithmetic operators

ในภาษา C# มีตัวดำเนินการทางคณิตศาสตร์เพื่อใช้ในการดำเนินการในคณิตศาสตร์ต่างๆ เช่น การบวก การลบ การคูณ การหาร และอื่นๆ ในการเขียนโปรแกรม มักเรียกตัวดำเนินการเหล่านี้ว่า operand เพื่อใช่จัดการกับข้อมูลที่มี

2.2.7.3 Compound assignment

Compound assignment operators เป็นตัวดำเนินการใช้สำหรับปรับปรุงหรืออัพเดทค่าของตัวแปร โดยมีข้อกำหนดว่าข้อมูลใหม่นั้นต้องมีความสัมพันธ์กับข้อมูลเดิม เช่น เพิ่มค่าของ x ไปอีก 10 กล่าวอีกนัยหนึ่งมันเป็นรูปแบบสั้นของ arithmetic operators

2.2.7.4 Increment และ decrement

Increment และ decrement เป็นตัวดำเนินการที่ใช้สำหรับเพิ่มค่าและลดค่าตามลำดับ ผลลัพธ์คือค่าของตัวแปรจะเพิ่มขึ้นหรือลดลง 1 เท่านั้น ซึ่งในตัวดำเนินการ increment และ decrement ก็จะแบ่งออกเป็น 2 แบบคือ prefix และ postfix และเครื่องหมายที่ใช้ในตัวดำเนินการนี้คือ ++ และ --Prefix เป็นการใส่เครื่องหมายไว้หน้าของตัวแปร การทำงานค่าของตัวแปรจะเพิ่มก่อนที่จะทำงานคำสั่งปัจจุบัน ในขณะที่ postfix เป็นการใส่เครื่องหมายทางด้านหลัง การทำงานค่าของตัวแปรจะถูกเพิ่มขึ้นหลังจากที่คำสั่งปัจจุบันทำงานเสร็จสิ้นแล้ว

2.2.7.5 Relational และ comparison operators

ตัวดำเนินการเปรียบเทียบ (comparison) และความสัมพันธ์ (relational) ถูกใช้ในการสร้าง expression กับตัวแปรและค่าคงที่ โดยผลลัพธ์ของ expression จะมีแค่เพียง true หรือ falseเท่านั้น มันมักจะใช้เพื่อสร้างเงื่อนไขเพื่อใช้กับคำสั่งในการเปรียบเทียบ เช่น if else, switch, while, do-while, for เป็นต้น

2.2.7.6 Logical operators

ตัวดำเนินการเชิงตรรกะ (logical operators) มักจะใช้ในการตรวจสอบกับค่าของ boolean โดยปกติตัวดำเนินการเปรียบเทียบจะให้ผลลัพธ์เป็น boolean อยู่แล้ว ในภาษา C# มีตัวดำเนินการเชิงตรรกะอยู่ 3 แบบคือ Not , And , Or

2.2.7.7 Conditional ternary operator (?)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ternary operator นั้นจะใช้เครื่องหมาย ? มันถูกใช้ในการประเมินค่าของ expression และจะให้ผลลัพธ์เป็นค่าแรกถ้าหาก expression เป็น true และค่าที่สองถ้า expression เป็น false โดยค่าทั้งสองต้องเป็นประเภทหรือชนิดข้อมูลชนิดเดียวกัน โดยมันมีรูปแบบการใช้งานดังนี้

```
expression ? value1 : value2;
String cc = "NY";
Console.WriteLine(cc == "NY" ? "New York" : "Other states");
```

ในตัวอย่าง จากคำสั่ง cc == "NY" นั้นคือ expression ที่สร้างขึ้นจาก ternary operator และ "New York" นั้นจะผลเป็นผลลัพธ์เมื่อ expression นี้เป็น true และค่าที่สองคือ "Other states" จะเป็นผลลัพธ์เมื่อ expression เป็น false โปรดจำไว้ว่าค่าทั้งสองจะต้องเป็นประเภทข้อมูลเดียวกันเสมอ

New York เป็นผลลัพธ์ของโปรแกรม มันจะแสดง "New York" เพราะว่า expression เป็น true

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนาแอปพลิเคชั่น

3.1 ความต้องการของระบบ

ระบบควบคุมและดูแลการเดินรถจำลองถูกพัฒนาขึ้นเพื่อช่วยให้ผู้ใช้งานพัฒนาทักษะการแก้ไขปัญหาที่เกิดขึ้นระหว่างการเดินรถตามตารางเวลาเดินรถ โดยที่พยายามให้มีการเดินรถเป็นไปตามที่กำหนดไว้ในตารางเดินรถให้ได้มากที่สุด และถ้ามีปัญหาเกิดขึ้นในระหว่างการเดินรถจะต้องพยายามควบคุมให้มีเวลาคลาดเคลื่อนน้อยที่สุด โดยเลือกวิธีแก้ปัญหาที่เหมาะสมกับสถานการณ์ เช่น การสับราง ปิดกั้นราง จองเส้นทางเดินรถ รวมถึงการปรับเปลี่ยนตารางเวลาการเดินรถ โดยแต่ละวิธีการจะทำให้ ระยะเวลา ระยะทาง และการใช้ทรัพยากรเชื้อเพลิงที่ไม่เท่ากันเป็นอันถือเป็นพารามิเตอร์ค่าใช้จ่าย (cost parameters) เมื่อการจำลองสิ้นสุดลง จะมีการประเมินผลการแก้ไขปัญหาออกมาโดยการรวมค่าใช้จ่าย(cost)ทั้งหมด นอกจากนี้ยังมีการแสดงค่าสถิติต่างๆ ระหว่างการจำลอง เช่น ค่าใช้จ่ายทั้งหมดที่เกิดขึ้น ความเร็วรถ เวลาที่คลาดเคลื่อน การแจ้งเตือนของปัญหาที่เกิดระหว่างการเดินรถ ตารางเวลารถไฟก่อนการเปลี่ยนแปลง ตารางรถไฟหลังการเปลี่ยนแปลง

3.2 การออกแบบโครงสร้างของแอปพลิเคชั่น

ผู้พัฒนาได้ศึกษาค้นคว้าทฤษฎีและหลักการพัฒนาแอปพลิเคชั่นโดยใช้เทคโนโลยีของซีชาร์ป และการนำยูนิตี้เกมเอนจินมาใช้รวมถึงการวางแผนและออกแบบซอฟต์แวร์ตามแบบของวิศวกรรมซอฟต์แวร์ ทั้งนี้เพื่อที่จะได้นำความรู้ที่ได้จากการศึกษาค้นคว้ามาพัฒนาซอฟต์แวร์ซึ่งเป็นแอปพลิเคชั่นระบบควบคุมและดูแลการเดินรถจำลอง

การพัฒนาระบบควบคุมและดูแลการเดินรถจำลองในครั้งนี้ ได้ทำการรวบรวมข้อมูลต่างๆ ทั้งทฤษฎีที่จำเป็นในการพัฒนาแอปพลิเคชั่น มีการค้นคว้าหาข้อมูลที่เกี่ยวข้อง ซึ่งผู้พัฒนาได้นำข้อมูลเหล่านี้มาใช้ในการออกแบบแอปพลิเคชั่น

โดยมีรายละเอียดของการออกแบบดังนี้

3.2.1 ลักษณะของแอปพลิเคชั่น

ระบบควบคุมและดูแลการเดินรถจำลองนี้มีลักษณะเป็นแอปพลิเคชัน โดยมีการแบ่งเป็น 2 ส่วนคือ

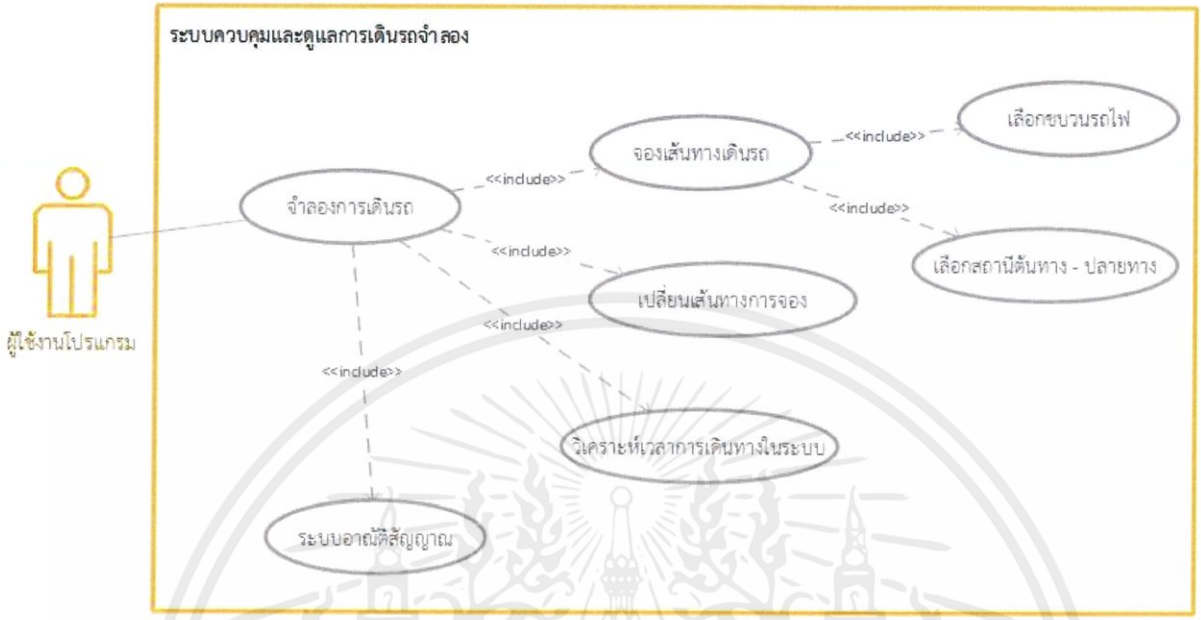
1.) Front-end จะถูกใช้งานโดยผู้ซึ่งจะเป็นผู้ควบคุมการเดินรถ โดยหน้าต่างการโต้ตอบระหว่างผู้ใช้พร้อมการแสดงค่าทางสถิติต่างๆ

2.) Back-end เตรียมข้อมูลที่ต้องใช้ในการคำนวณในต้นเริ่มต้นโปรแกรมหลังจากที่ได้รับค่าต่างๆ ที่จำเป็นจากการใช้งานของผู้ใช้แล้ว back end จะทำการประมวลผลข้อมูลต่างๆ และเก็บผลในการใช้งานของผู้ใช้ เช่น เวลาล่าช้า และทำการส่งข้อมูลไปถึง Front-end เพื่อแสดงผลลัพธ์ที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.1 ยูสเคส (Use Case)

ยูสเคสไดอะแกรม (Use Case Diagram)



รูปภาพที่ 3.1 ยูสเคสไดอะแกรม (Use Case Diagram)

จากไดอะแกรมข้างต้นแสดงให้เห็นว่าโปรแกรมมี Use Case ทั้งหมด 8 กรณีดังต่อไปนี้

1.) Use Case Diagram ระบบจำลองการเดินรถ

ตารางที่ 3.1 แสดงรายละเอียด Use Case ระบบจำลองการเดินรถ

Use Case ID	1
Use Case Name	เฝ้าดูระบบจำลองการเดินรถ
Actor	ผู้ใช้งานโปรแกรม
Description	ผู้ใช้ต้องการเฝ้าดูการทำงานของระบบจำลองการเดินรถ ซึ่งจะต้องเริ่มการจำลองระบบก่อน โดยในโปรแกรมจะต้องแสดงเส้นทางรางทั้งหมดและขบวนรถไฟที่ได้ทำการเลือกเส้นทางเดินรถให้แล้วเรียบร้อยแล้ว และมีตารางแสดงเวลาที่รถไฟควรจะเดินทางถึงสถานี และตารางแสดงเวลาเมื่อขบวนรถไฟที่ทำการจำลองถึงสถานีทุกครั้ง
Precondition	1. ผู้ใช้จองเส้นทางการเดินรถ 2. ผู้ใช้เริ่มการจำลองการเดินรถ
Postcondition	1. ระบบจำลองจะเริ่มการทำงานตามที่ผู้ใช้งานกำหนดไว้ 2. ระบบแจ้งเตือนจะเริ่มการทำงานในการแสดงรายละเอียดการจำลองการเดินรถ และแสดงเวลาการเดินรถของขบวนรถไฟเพื่อเดินทางถึงสถานี
Normal Flow	1. ผู้ใช้สามารถสังเกตุการทำงานของระบบหลังจากกดปุ่มเริ่มการทำงานระบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	2. โปรแกรมจะเริ่มทำการจำลองโดยจะแสดงผลเส้นทางเดินรถ พร้อมแสดงรายละเอียดข้อมูลขบวนรถที่กำลังจำลอง แสดงตารางเวลาที่ขบวนรถที่จำลองควรเดินทางถึงในแต่ละสถานี และตารางที่แสดงเวลาที่ขบวนรถจำลองเดินทางถึงสถานีในระบบจำลอง 3. ระบบทำการจำลองต่อจนเสร็จ
Extension Flow	Case เมื่อเกิดรางขัดข้องในระหว่างเฝ้าดูการจำลอง 1. ระบบทำการแจ้งเตือนให้ผู้ใช้เห็นว่ารางนั้นขัดข้อง 2. ผู้ทำการเปลี่ยนเส้นทางการจองโดยที่จะต้องให้ขบวนรถไฟสามารถเดินต่อได้จนถึงสถานีต่อไปที่กำหนดไว้ 3. ระบบทำการจำลองต่อจนเสร็จ
Include	การจองเส้นทางเดินรถ
Extend	การเปลี่ยนเส้นทางการจอง

2.) Use Case Diagram ระบบอาณัติสัญญาณ

ตารางที่ 3.2 แสดงรายละเอียด Use Case Diagram ระบบอาณัติสัญญาณ

Use Case ID	2
Use Case Name	ระบบอาณัติสัญญาณ
Actor	ผู้ใช้งานโปรแกรม
Description	ผู้ใช้ต้องการเปลี่ยนสัญญาณในระบบจำลองสามารถทำได้โดยการคลิกไปป้ายสัญญาณตัวนั้นๆ ซึ่งมีการควบคุมได้ทั้งหมด 2 แบบ คือ - สีเขียว : ขบวนรถไฟสามารถวิ่งผ่านได้ - สีแดง : รถไฟต้องหยุดจอดรอตามป้ายบอกอาณัติสัญญาณ แต่ควรระวังไว้ว่าป้ายบอกอาณัติสัญญาณจะสามาถแจ้งให้ขบวนรถไฟให้ทำตามได้นั้นรถไฟต้องอยู่ในระยะที่สามารถเห็ยป้ายอาณัติสัญญาณด้วย
Precondition	เมื่อผู้ใช้เปิดโปรแกรมระบบจำลอง
Postcondition	การแสดงผลของป้ายอาณัติสัญญาณที่ถูกคลิกเปลี่ยน
Normal Flow	1. ผู้ใช้ทำการเลือกป้ายอาณัติสัญญาณที่ต้องการเปลี่ยนแล้วคลิกที่ป้ายอาณัติสัญญาณนั้น 2. โปรแกรมจะทำการเปลี่ยนสัญญาณให้โดยอัตโนมัติ

3.) Use Case Diagram จองเส้นทางเดินรถ

ตารางที่ 3.3 แสดงรายละเอียด Use Case Diagram จองเส้นทางเดินรถ

Use Case ID	UC03
Use Case Name	จองเส้นทางเดินรถ
Actor	ผู้ใช้งานโปรแกรม
Description	ผู้ใช้ต้องการจองเส้นทางเดินรถ จะต้องมีการกำหนดขบวนรถที่จะใช้ในการจำลองอย่างน้อยหนึ่งขบวน และต้องกำหนดเส้นทางการเดินรถโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	กำหนดสถานีต้นทางปลายทาง ถึงจะสามารถกดเริ่มระบบจำลองได้
Precondition	1. ผู้ใช้ต้องทำการเลือกขบวนรถที่ต้องการจำลอง 2. ผู้ต้องใช้เลือกเส้นทางการเดินรถโดยเลือกสถานีต้นทางและสถานีปลายทาง
Postcondition	1. ระบบเริ่มการจำลองการเดินรถ 2. ระบบจับเวลาการจำลองการเดินรถเริ่มทำงาน
Normal Flow	1. ผู้ใช้ทำการเลือกขบวนรถที่ต้องการจำลอง 2. ผู้ใช้เลือกสถานีต้นทางและสถานีปลายทาง 3. ผู้ใช้กดปุ่มเริ่มการจำลองการเดินรถ
Include	- การเลือกขบวนรถไฟ - การเลือกสถานีต้นทางและปลายทาง

4.) Use Case Diagram การเปลี่ยนเส้นทางการจอง

ตารางที่ 3.4 แสดงรายละเอียด Use Case Diagram การเปลี่ยนเส้นทางการจอง

Use Case ID	4
Use Case Name	เปลี่ยนเส้นทางการจอง
Actor	ผู้ใช้งานโปรแกรม
Description	ผู้ใช้ต้องการเปลี่ยนเส้นทางการเดินรถจากเส้นทางเดิมที่ขบวนรถจำลองได้ทำการจองไว้โดยผู้ใช้ ซึ่งจะต้องเป็นเส้นทางที่รถไฟยังไม่เดินทางผ่าน และต้องไม่เป็นรางที่รถไฟกำลังวิ่งอยู่ โดยผู้ใช้สามารถกดไปที่รางที่ต้องการเปลี่ยนเพื่อเปลี่ยนสถานะของรางนั้นๆ ได้
Precondition	1. ผู้ใช้เริ่มการจำลองการเดินรถ 2. รางที่ต้องการเปลี่ยนสถานะการจองต้องไม่มีขบวนรถวิ่งอยู่
Postcondition	สถานะของรางเปลี่ยนไปตามที่ผู้ใช้กำหนด
Normal Flow	1. ผู้ใช้ทำการคลิกไปทางรางที่ต้องการเปลี่ยนสถานะ 2. รางถูกเปลี่ยนสถานะการจองตามที่ผู้ใช้กำหนด
Extension Flow	Case รางไม่สามารถใช้งานได้เนื่องจากเกิดเหตุการณ์ผิดปกติกับรางนั้น 1. ทำการแจ้งเหตุขัดข้องให้แก่ผู้ใช้เพื่อทำการแก้ไขปัญหา 2. ผู้ใช้ทำการคลิกไปทางรางที่ต้องการเปลี่ยนสถานะโดยเลี่ยงรางที่ไม่สามารถใช้งานได้ 3. รางถูกเปลี่ยนสถานะการจองตามที่ผู้ใช้กำหนด

5). Use Case Diagram วิเคราะห์เวลาการเดินทางในระบบ

ตารางที่ 3.5 แสดงรายละเอียด Use Case Diagram วิเคราะห์เวลาการเดินทางในระบบ

Use Case ID	5
Use Case Name	วิเคราะห์เวลาการเดินทางในระบบ
Actor	ผู้ใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Description	ผู้ใช้ต้องการวิเคราะห์เวลาจากจำลองที่ได้นั้นว่ามีประสิทธิภาพใกล้เคียงกับเวลาที่ผู้ใช้โปรแกรมต้องการหรือไม่ โดยระบบจำลองนั้นมีแสดงเวลาที่ผู้ใช้ต้องการในโปรแกรม และเวลาที่ระบบจำลองคำนวณออกมา
Precondition	ผู้ใช้เริ่มระบบจำลองแล้ว
Postcondition	ระบบจำลองแสดงเวลาที่ขบวนรถแต่ละขบวนที่กำลังจำลองอยู่ออกมา และจะมีการแจ้งทุกครั้งเมื่อเดินทางถึงสถานี โดยจะแสดงชื่อสถานีและเวลาที่ขบวนรถไฟถึงสถานี
Normal Flow	1. ขบวนรถไฟเดินทางถึงสถานี 2. ระบบจำลองแจ้งเตือนให้ผู้ใช้งานเห็นโดยแสดงแสดงชื่อสถานีและเวลาที่ขบวนรถไฟถึงสถานี

6.) Use Case Diagram เลือกขบวนรถไฟ

ตารางที่ 3.6 แสดงรายละเอียด Use Case Diagram เลือกขบวนรถไฟ

Use Case ID	6
Use Case Name	เลือกขบวนรถไฟ
Actor	ผู้ใช้งานโปรแกรม
Description	ผู้ใช้ต้องการเลือกขบวนรถไฟ ซึ่งสามารถเลือกขบวนรถที่ต้องการมาใช้ในการจำลองได้
Precondition	เมื่อผู้ใช้เปิดโปรแกรมระบบจำลอง
Postcondition	ขบวนรถที่ถูกเลือกจะถูกนำมาใช้เมื่อเริ่มระบบจำลองการเดินรถ
Normal Flow	1. ผู้ใช้เปิดโปรแกรม 2. ผู้ใช้เลือกขบวนรถที่ต้องการมาใช้โดยจะมีเมนูจากด้านหน้าโปรแกรม

7.) Use Case Diagram เลือกสถานีต้นทาง - ปลายทาง

ตารางที่ 3.7 แสดงรายละเอียด Use Case Diagram เลือกสถานีต้นทาง - ปลายทาง

Use Case ID	7
Use Case Name	เลือกสถานีต้นทาง - ปลายทาง
Actor	ผู้ใช้งานโปรแกรม
Description	ผู้ใช้ต้องการเลือกสถานีต้นทางและปลายทางให้กับขบวนรถไฟที่ถูกเลือกไว้
Precondition	ผู้ใช้ได้เลือกขบวนรถแล้ว
Postcondition	ขบวนรถจะทำการจองรางโดยมีขอบเขตจากสถานีต้นทางถึงสถานีปลายทาง
Normal Flow	1. ผู้ใช้ไปที่เมนูเลือกสถานีต้นทางและปลายทาง 2. ผู้ใช้เลือกสถานีต้นทางและปลายทางที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	3. ขบวนรถที่ถูกเลือกจะทำการจองรางโดยมีขอบเขตจากสถานีต้นทางถึงสถานีปลายทาง

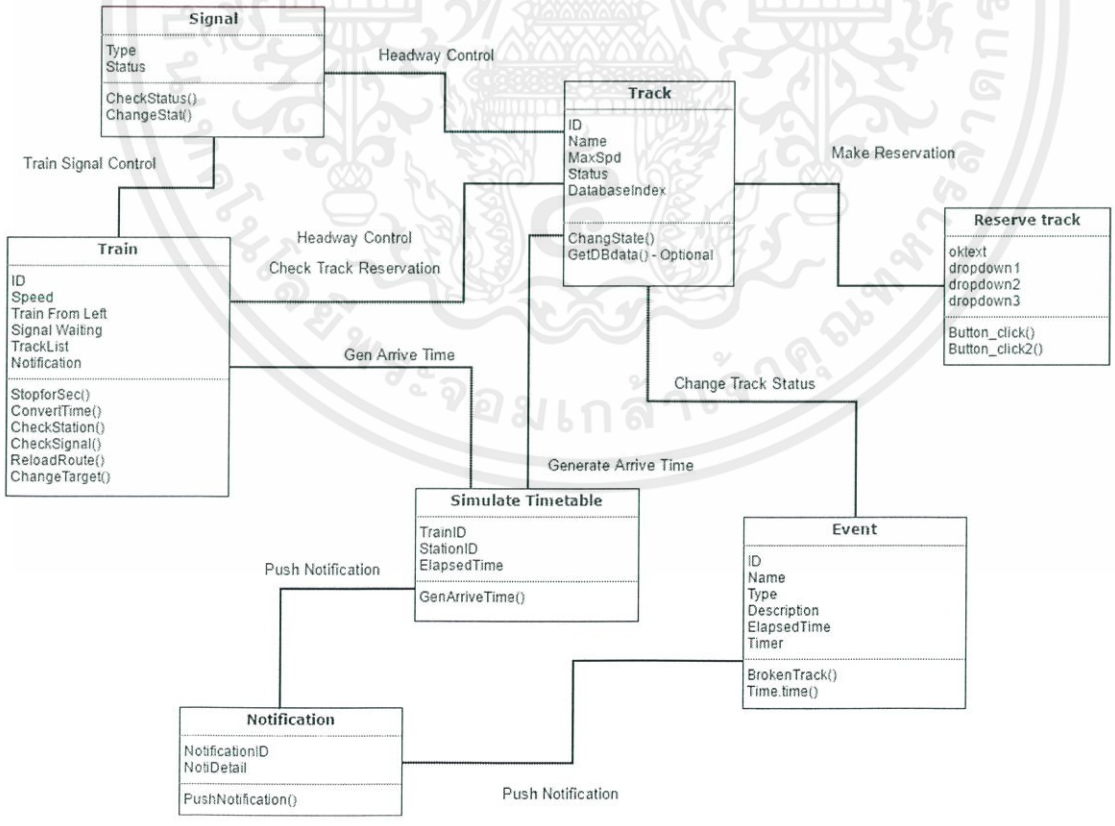
3.2.2 ส่วนประกอบของระบบจำลอง

จากความต้องการของระบบและลักษณะของ Simulator ที่แสดงอยู่ใน Use Case ข้างต้น ผู้พัฒนาสามารถแบ่ง Simulator ออกเป็นระบบย่อยๆ ได้ดังนี้

- Block Track
- Switch Track
- Release Track
- Signaling
- Reserve Track
- Timetable Management
- Time Calculation
- Event

3.2.3 คลาสไดอะแกรม (Class Diagram)

จากความสัมพันธ์ของ Use Case ข้างต้น ในหัวข้อ 3.2.2 เราจะสามารถนำมาแบ่งเป็นคลาสไดอะแกรม (Class Diagram) ได้ดังนี้



รูปภาพที่ 3.2 คลาสไดอะแกรม (Class Diagram)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 Meta Data

Station

Field	Data Type	Comment
Station_ID	INT(5)	ไอดีสถานี
Station_Cost	INT(3)	ค่าเวลาที่ใช้ในการจอดสถานี
Station_Name	Varchar(20)	ชื่อสถานี
Station_Status	Varchar(10)	เก็บสถานะของสถานีว่ามีรถไฟจอดอยู่หรือไม่

ตารางที่ 3.8 metadata station

Train

Field	Data Type	Comment
Train_ID	INT(5)	ไอดีของขบวนรถ
Train_Type	Varchar(10)	ประเภทของขบวนรถ
Track_list	Array List	เก็บค่ารางที่ถูกจองเพื่อรถไฟคันนี้
Speed	INT(2)	สปีดปัจจุบันของรถไฟ

ตารางที่ 3.9 metadata train

Track

Field	Data Type	Comment
Max Speed	INT(2)	Track maxspd
Track_ID	INT(5)	ชื่อราง
Track_Type	Varchar(10)	ประเภทของราง
Track_Status	Varchar(10)	สถานะราง ว่าง หรือถูกจองโดยรถขบวนไหน

ตารางที่ 3.10 metadata track

Signal Control

Field	Data Type	Comment
Signal_ID	INT(5)	ไอดีของตัวสัญญาณ
Switch_Status	INT(1)	สถานะของตัวสัญญาณว่าเป็
Switch_Type	Varchar(10)	ประเภทของตัวสับรางก

ตารางที่ 3.11 metadata switch control

Time Calculator

Field	Data Type	Comment
Train_ID	INT(5)	ไอดีรถไฟ
Station_ID	INT(5)	ไอดีสถานี
Station_Arrive_Time	Varchar(5)	เวลาที่รถไฟถึงสถานี

ตารางที่ 3.12 metadata score cal

Event

Field	Data Type	Comment
Event_ID	INT(5)	ไอดีของเหตุการณ์เกิดขึ้นในการทดสอบ
Event_Type	Varchar(10)	ประเภทของเหตุการณ์ที่เกิด
Event_Detail	Varchar(100)	รายละเอียดของเหตุการณ์ที่ต้องแจ้งผู้ใช้งาน
Event_Condition	Varchar(100)	เงื่อนไขของเหตุการณ์ว่าถ้าเกิดเหตุการณ์นี้แล้วจะมีอะไรเกิดขึ้นบ้าง

ตารางที่ 3.13 metadata event

Notification

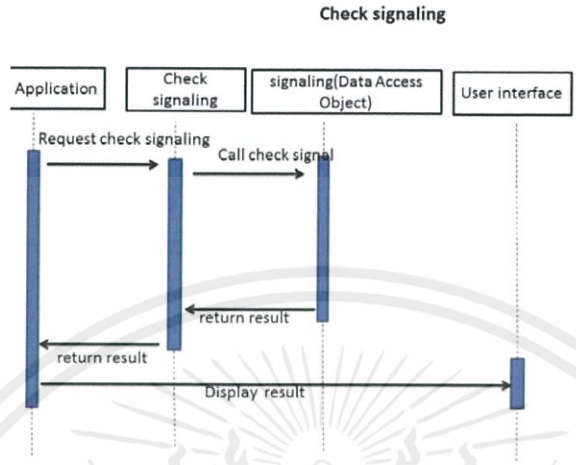
Field	Data Type	Comment
Notification_ID	INT(5)	ไอดีของการแจ้งเตือน
Notification_Type	Varchar(10)	ประเภทการแจ้งเตือน
Notification_Detail	Varchar(100)	รายละเอียดของการแจ้งเตือน
Notification_Condition	Varchar(100)	เงื่อนไขของเหตุการณ์ว่าถ้าแจ้งเตือนแล้วจะมีอะไรเกิดขึ้นบ้าง

ตารางที่ 3.14 metadata notification

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

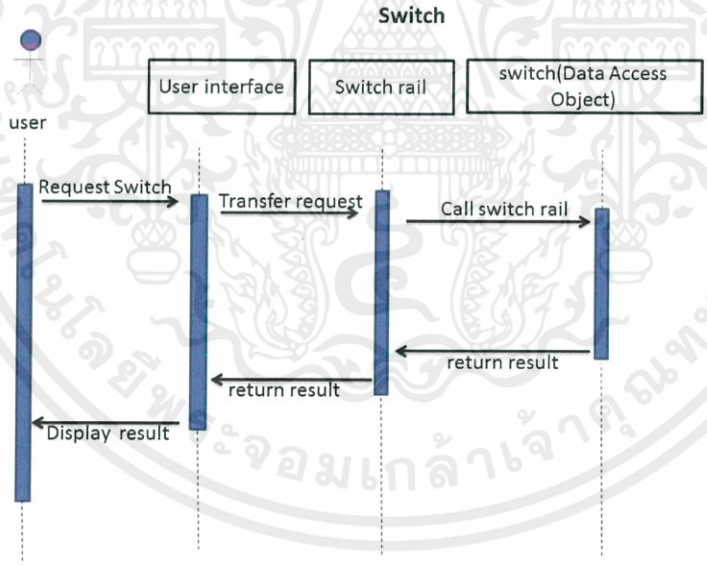
3.3 Sequence Diagram

3.3.1 Check Signal



รูปที่ 3.3 Check Signal Sequence Diagram

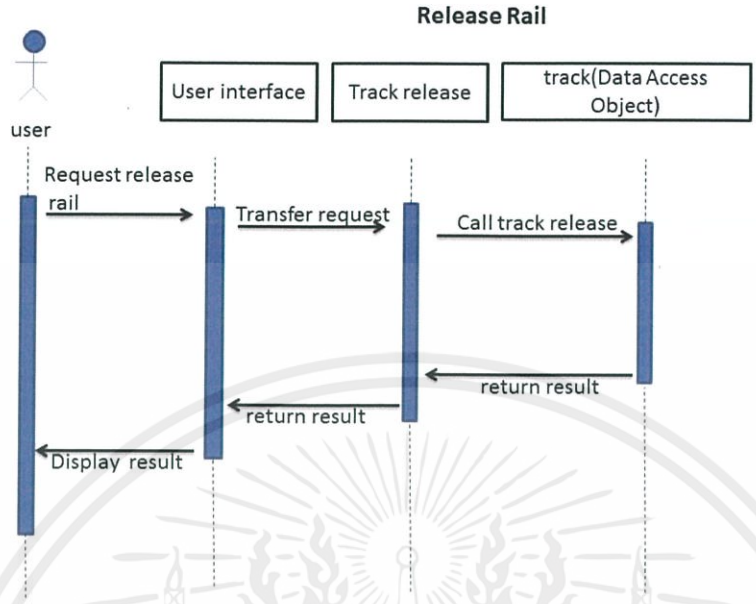
3.3.2 Switch



รูปที่ 3.4 Switch Sequence Diagram

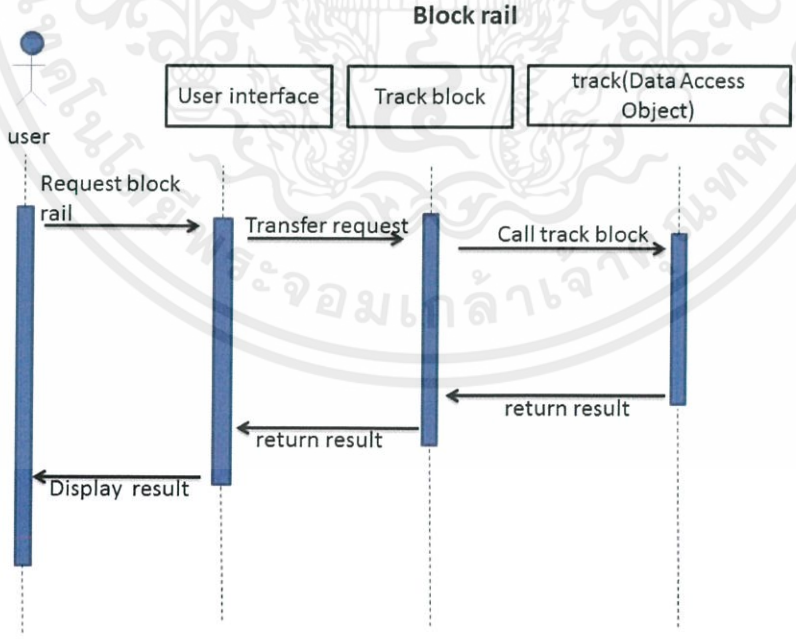
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 Release Rail



รูปภาพที่ 3.5 Release Rail Sequence Diagram

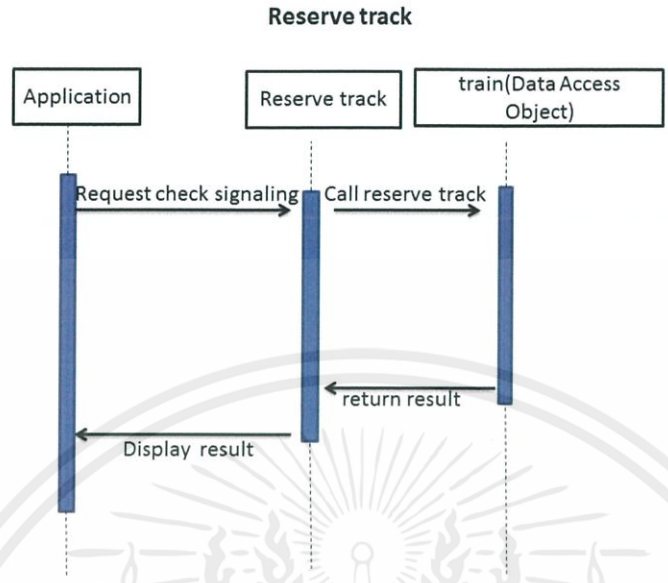
3.3.4 Block Rail



รูปภาพที่ 3.6 Block Rail Sequence Diagram

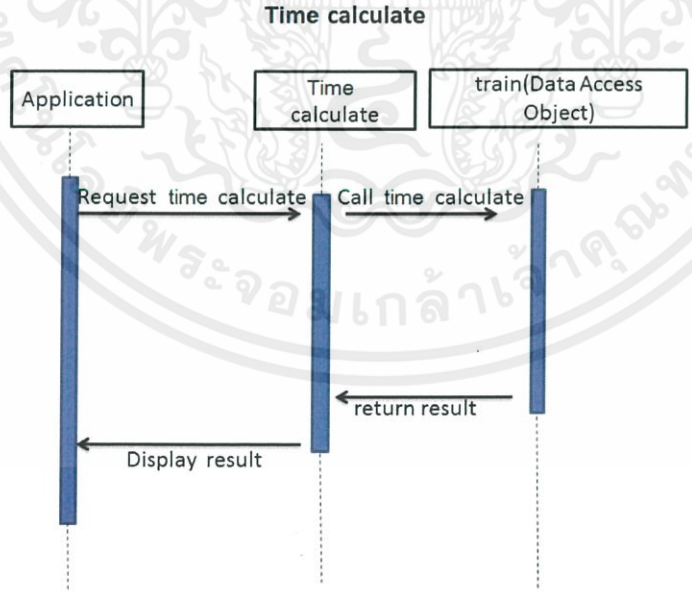
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 Reverse Track



รูปภาพที่ 3.7 Reverse Track Sequence Diagram

3.3.6 Time Calculator



รูปภาพที่ 3.8 Score Calculator Sequence Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การพัฒนาโปรแกรม

3.4.1 การสร้างรางใหม่

ในการสร้างรางสำหรับระบบจำลองการเดินรถใหม่นั้นจะมีขั้นตอนดังนี้

- ทำการออกแบบรางที่จะเป็นส่วนต่อขยาย
- ทำการวัดระยะ และคำนวณขนาดของ Sprite โดยจะคำนวณจากระยะทางจริง ในหน่วยกิโลเมตร คูณด้วย 1.5 เท่า จะได้ความยาวรางเป็นขนาดของ Sprite รางในแกน x
- จัดตำแหน่ง Sprite ให้เข้าที่ และ ทำการเช็คค่าต่างๆ ของระบบ จากนั้นนำสคริป Track ลากมาใส่ยัง Sprite ที่ได้
- กำหนดค่าของราง อาทิเช่น ไอดี ความเร็ว ชนิดของราง โดยรางแต่ละชนิดจะมี Max Speed ที่ไม่เท่ากัน ซึ่งถ้าหากเลือกทำในข้อนี้แล้ว สามารถข้ามขั้นตอนในขั้นตอนต่อไปได้ทันที
- ใส่ค่าพารามิเตอร์ของรางในดาต้าเบส และใส่ Database Index ที่ได้ในพารามิเตอร์ของราง ในกรณีที่ใช้การดึงข้อมูลจากดาต้าเบส (Optional)
- ทำการกำหนดค่า Object name ของรางทั้งหมดใหม่ โดยการตั้งชื่อนั้น จำเป็นจะต้องอ้างอิงจากหัวข้อ 4.x.x เพื่อให้สอดคล้องกลับอัลกอริธึมของรถไฟ
- กำหนด Status เริ่มต้นของรางให้เป็น active เพื่อให้รางเป็นสถานะว่าง ยังไม่ถูกจอง และกรณีเป็นรางที่เป็นชานชลาของสถานี ให้กำหนดค่า ID เป็นชื่อย่อของสถานีเสมอ

3.4.2 การสร้างรถไฟใหม่

ในการสร้างรถสำหรับระบบจำลองการเดินรถใหม่นั้น จะมีขั้นตอนดังนี้

- ทำการเพิ่มสไปรต์ของรถไฟ โดยอาจจะใช้ของรถไฟคันเดิมแล้วนำมาแก้ไข
- ทำการคัดลอกสคริป Train ของรถไฟเอาไว้ แล้วนำมาใส่ใน Object ของรถไฟที่เพิ่งสร้างใหม่ โดยทำการแก้ไขโค้ดในส่วนของชื่อรถไฟภายในสคริป และทำการเปลี่ยนชื่อสคริปใหม่เป็น Train ตามด้วยลำดับของรถไฟ
- ทำการกำหนดค่าของรถไฟ โดยกำหนดค่า Max Speed ของรถไฟ และกำหนดทิศทางของรถไฟหากรถไฟมาจากทางขวา ให้ทำการติ๊ก Train From Right เพื่อปรับอัลกอริทึ่มของรถไฟ

3.4.3 การสร้างเสาสัญญาณใหม่

- ทำการคัดลอก Sprite ของเสาสัญญาณในประเภทเดียวกันมา โดยเสาสัญญาณจะแบ่งได้เป็นทั้งหมด 3 แบบ คือ Two-way signal , One-way (To Right) Signal และ One-way (To Left) Signal
- ทำการกำหนด Type ของสัญญาณ โดย Type จะทำการเรียงตามประเภทของเสาสัญญาณดังนี้
  - Two-way signal

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- One-way (To Right) Signal
- One-way (To Left) Signal

- ทำการกำหนดค่า Status ของเสาสัญญาณ โดยค่า Status จะมีค่าต่างกันไปในแต่ละ Type ของ Signal ซึ่งสามารถเขียนได้เป็น Case ดังนี้
  - Type 1 , Status 0 : รถไฟจากทางซ้ายหยุด รถไฟจากทางขวาผ่าน
  - Type 1 , Status 1 : รถไฟจากทางซ้ายผ่าน รถไฟจากทางขวาหยุด
  - Type 2 , Status 0 : รถไฟจากทางซ้ายหยุด
  - Type 2 , Status 1 : รถไฟจากทางซ้ายผ่าน
  - Type 3 , Status 0 : รถไฟจากทางขวาผ่าน
  - Type 3 , Status 1 : รถไฟจากทางขวาหยุด

- ทำการปรับตำแหน่งของ Collider ทางด้านซ้าย และด้านขวา ให้อยู่ขอบของราง ติดกับเสาสัญญาณอีกต้น ตามรูปที่ 4.3.3.3 Track Collider & Headway
- หากเป็นส่วนที่มีการสับรางเกิดขึ้น จะต้องกำหนดตำแหน่งของ Collider ที่อยู่ขอบซ้ายและขวาให้ครอบคลุมรางทั้งสองเส้น เพื่อให้ครอบคลุมทั้งสองกรณี

3.4.4 การสร้าง Event ใหม่

- ทำการออกแบบอีเวนท์ใหม่ โดยคำนึงถึงเหตุการณ์ที่จะเกิด อุปกรณ์ที่จะเสีย หรือรางที่จะเกิดปัญหา
- ทำการออกแบบช่วงเวลาที่จะเกิด Event ขึ้น โดยอาจจะใช้การกำหนดเวลาโดยตรงว่าจะให้เกิดหลังผ่านไปกี่นาที หรือจะใช้ฟังชั่น Random() ของ C# เพื่อสุ่มเวลาที่จะเกิดขึ้นของ Event ก็ได้เช่นกัน
- ทำการเขียนแก้ไขสคริป Event เพื่อทำการเพิ่ม Event ที่จะเกิดขึ้น โดยสามารถเขียนได้หลายวิธีขึ้นอยู่กับผู้ออกแบบว่าต้องการจะให้ Event ออกมาแบบใด โดยสามารถดูตัวอย่างโค้ดของรางเสีย ได้จากรูป 4.3.5.3

3.4.5 การสร้างระบบการจองราง

- ให้เริ่มจากดูว่าสถานีต้นทางและปลายทางที่เราต้องการทำระบบจองรางนั้นคือสสถานีอะไรและในเส้นทางนี้มีรางอะไรอยู่ในเส้นทางนี้บ้างเนื่องจากเราต้องใช้หมายเลขของแต่ละรางในการทำระบบจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ให้ทำการแก้ไขสคริป reservetrain โดยใช้ชื่อของรางทั้งหมดในเส้นทางที่เราต้องการมาเขียนเงื่อนไขเมื่อได้รับค่าจากดรอปดาวน์อินพุตทั้ง2จากผู้ใช้
- โดยเริ่มจากกาตรวจสอบค่าสถานะในรางทั้งหมดว่ามีสถานะเป็น active อยู่หรือไม่
- จากนั้นให้เรากำหนดเงื่อนไขว่าถ้าสถานะของรางนั้นเป็น active อยู่ให้แสดงคำว่า click reserve เพื่อให้รู้ว่าสามารถทำการจองรางได้
- ถ้าสถานะของรางทั้งหมดนั้นไม่ได้เป็น active อยู่ให้แสดงคำว่า track bust แทนเพื่อให้รู้ว่าไม่สามารถทำการจองรางได้
- จากนั้นให้เราใส่สคริปให้กับปุ่ม reserve ถ้าสถานะของรางนั้นเป็น active ทั้งหมดให้เปลี่ยนสถานะของรางตามขบวนรถที่เราเลือกไว้เมื่อปุ่มถูกกด

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 การเข้าใช้งาน

การทำงานของโปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถจะประกอบด้วยส่วนของผู้ใช้งานที่ต้องทำหน้าที่ตัดสินใจและออกคำสั่งต่างๆผ่าน User Interface ของ โปรแกรมและอีกส่วนหนึ่งคือส่วนของโปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถที่จะต้องทำหน้าที่รับคำสั่งจากผู้ใช้งานและสร้างสถานการณ์ต่างๆ ซึ่งระบบจะทำการจำลองการเดินรถ และเก็บสถิติการเดินรถเพื่อเปรียบเทียบ Ideal Timetable ที่กำหนดไว้ล่วงหน้ากับ Simulate Timetable ที่ผู้ใช้งานได้ทำการทดลองควบคุมการเดินรถ

4.1.1 การเข้าใช้งานโปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถ

ในการเข้าใช้งานโปรแกรมจำลองการควบคุมรถจากแผ่น CD ที่ได้รับ ในโฟลเดอร์ของโปรแกรม จะพบกับโปรแกรมทั้งสองเวอร์ชันแยกกันอยู่ใน Folder 1440x900 และ 1920x1080 โดยในแต่ละโฟลเดอร์เมื่อเข้าไปจะพบกับโปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถ

- **W1440x900.exe สำหรับการตั้งค่าหน้าจอที่มีความระเอียด 1440x900**
- **W1920x1080.exe สำหรับการตั้งค่าหน้าจอที่มีความระเอียด 1920x1080**

ซึ่งทั้งสองเวอร์ชันได้ถูกออกแบบมาใช้งานกับความละเอียดที่กำหนดไว้ โดยเมื่อดับเบิ้ลคลิกจะพบกับหน้าสำหรับตั้งค่าการทำงาน การตั้งค่าการทำงานเบื้องต้นเบื้องต้นโดยเลือกการตั้งค่าหน้าจอที่มีความระเอียด ให้ตรงกับที่กำหนดไว้ในชื่อโปรแกรม

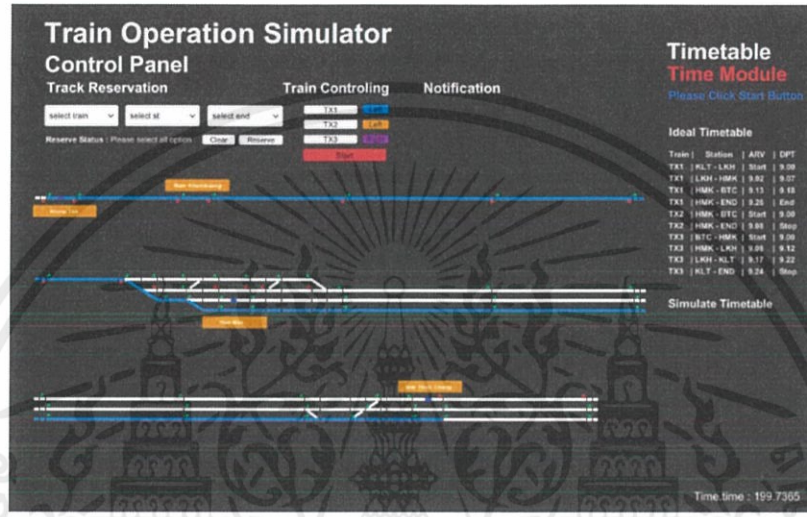
รูปที่ 4.1 ตัวอย่างหน้าจอ การตั้งค่าการทำงานเบื้องต้นของ Unity

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 โปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถ

ในส่วนของ User Interface ของโปรแกรม จะแบ่งได้เป็น 3 ส่วนหลักๆ คือ

- Control Panel
- Timetable
- Simulator



รูปที่ 4.2 ตัวอย่างโปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถ

4.1.2.1 Control Panel

เป็นส่วนที่ใช้ในการตั้งค่าเบื้องต้นของ Track และการควบคุมเบื้องต้นของโปรแกรม โดยจะแบ่งเป็น

- Track Reservation จะเป็นระบบการจองรางแบบอัตโนมัติ ซึ่งจะกล่าวถึงในบทต่อๆไป
- Train Controlling จะคอยบอกว่ารถไฟคันใดมีทิศทางไปทางใด และใช้เส้นทางสีอะไร
- Notification เป็นส่วนแจ้งเตือน เมื่อเกิดเหตุการณ์ขึ้นและแจ้งเตือนว่าควรจะทำอะไรต่อไป

4.1.2.2 Timetable

เป็นส่วนแสดงผลทางเวลาปัจจุบัน และตารางการเดินรถที่กำหนดเอาไว้ เวลาที่ใช้ในการเดินรถทั้งหมด รวมไปถึงตารางการเดินรถจริงที่ผู้ใช้ได้ทำการจำลองขึ้นมา โดยจะประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Time** แสดงค่าเวลาปัจจุบันในขณะที่รถเดินอยู่
- **Time Elapsed** แสดงจำนวนเวลาที่ผ่านไปเมื่อเริ่มกด Start
- **Ideal Timetable** แสดง Timetable ที่ถูกจัดไว้ ซึ่งเป็นเวลาในอุดมคติว่าไม่เกิดเหตุการณ์ผิดปกติขึ้น
- **Simulated Timetable** แสดง Timetable ที่ได้จากการทดลองใช้โปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถจริงของผู้ใช้งาน

รูปที่ 4.3 Timetable Interface

4.1.2.3 Simulator

เป็นส่วนที่ใช้แสดงผลการจำลองของโปรแกรม ซึ่งจะลงรายละเอียดในบท 4.3

4.2 การตั้งค่าก่อนการจำลองการเดินรถ

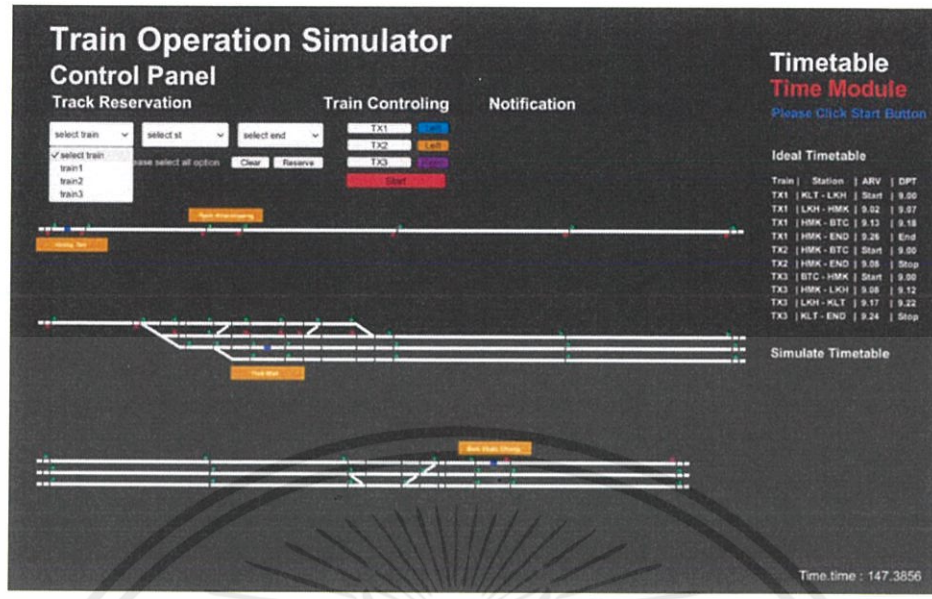
4.2.1 เลือกเส้นทางการเดินรถ

ก่อนเริ่มทำการ Simulate ให้ User ทำการศึกษาเส้นทางและเวลารถออกในแต่ละเส้นทางจากตารางการเดินรถ (timetable) ที่ทำการสร้างเอาไว้ใน Ideal Timetable เบื้องต้น เพื่อใช้ในการควบคุมรถไฟ

4.2.2 เริ่มการจำลองการควบคุมระบบการเดินรถ

เมื่อเริ่มทำการจำลองผู้ใช้จะต้องทำการจองรางของรถแต่ละคัน โดยใช้ Track Reservation Module หรือใช้การจองด้วยมือ โดยการดับเบิ้ลคลิกไปที่รางโดยตรง จนกว่าจะได้สถานะของการจองที่ต้องการ จากนั้น เมื่อทำการจองรางเสร็จสมบูรณ์ โดยตรวจสอบเส้นทางทั้งหมดแล้ว ให้ทำการกดปุ่ม Start เพื่อทำการเริ่มการจำลองโดยเวลาเริ่มต้นในระบบจำลองจะเป็น 8.55 ซึ่งเวลาเดินรถเริ่มต้นจะเป็น 9.00 ซึ่งจะมีเวลาให้ผู้ใช้งานเตรียมความพร้อม 5 นาที ก่อนรถไฟจะออกจากสถานี

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การจองรางด้วย Track Reservation

4.2.3 คำสั่งของผู้ใช้ต่อวัตถุต่างๆ ในโปรแกรมจำลองการควบคุมและดูแลการเดินรถ

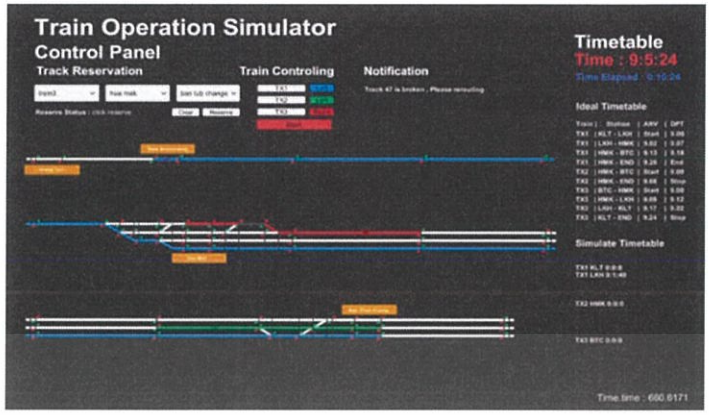
ระหว่างการควบคุมระบบจำลองรถผู้ควบคุมจะสามารถใช้คำสั่งในการสั่งการ และแก้ไขสถานการณ์ต่างๆ ได้ดังนี้

- การจองราง ทำได้โดยการดับเบิ้ลคลิกเพื่อทำการเปลี่ยนสถานะของการจองรางได้ โดยจะต้องทำก่อนรถไฟจะวิ่งผ่านใน 2 ช่วงสัญญาณไฟ
- การปรับสัญญาณไฟ ทำได้โดยการดับเบิ้ลคลิกที่ไฟสัญญาณ เพื่อทำการเปลี่ยนสถานะของเสาไฟ โดยจะต้องทำในระยะ 2 ช่วงสัญญาณไฟเช่นเดียวกัน โดยจะมีข้อยกในส่วนของสัญญาณไฟในช่วงของสถานี ซึ่งใช้ในกรณีต้องการปล่อยรถออกจากสถานีช้ากว่าปกติ
- การสับราง ปกติแล้วรางสับจะไม่ได้เชื่อมกับรางหลักจนกว่าจะมีการจองรางเพื่อให้รถไฟทำการสับรางเกิดขึ้น ซึ่งรายละเอียดในส่วนนี้สามารถอ่านเพิ่มเติมได้จากหัวข้อ 4.3.1.3

4.2.4 การตัดสินใจเมื่อเกิดเหตุการณ์ที่ไม่ปกติต่างๆ

เมื่อถึงจุดหนึ่งของการจำลองจะมีการสุ่มเหตุการณ์ต่างๆที่อาจเกิดขึ้นได้ขึ้นเพื่อดูการตัดสินใจของผู้ใช้งาน ซึ่งเมื่อเกิดเหตุการณ์ขึ้น ผู้ใช้อาจจะเลือกใช้สัญญาณไฟเพื่อหยุดรถไฟชั่วคราว และทำการเปลี่ยนเส้นทางก่อนจะเกิดอุบัติเหตุขึ้น ซึ่งการกระทำในจุดนี้จะขึ้นอยู่กับการตองสนองของผู้ควบคุมว่าควรจะทำการแก้ไขอย่างไรที่จะแก้ไขปัญหาเพื่อ ให้เกิดผลกระทบคนส่วนมากน้อยที่สุด

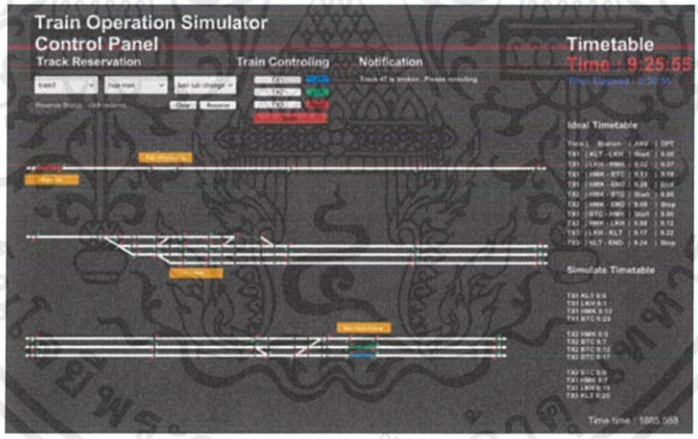
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 Track Broken Incident

4.2.5 การประเมินผู้ใช้โปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถ

การประเมินผลผู้ใช้งาน จะใช้การตรวจสอบจากเวลาของรถไฟที่ถึงสถานี โดยโปรแกรมจะมีการเก็บค่าในส่วนของเวลาที่รถไฟวิ่งมาถึงสถานี และบันทึกค่าไว้ใน Simulate Timetable ซึ่งเราสามารถนำค่านี้มาอ้างอิงในการประเมินผู้ใช้งาน โดยนำไปเปรียบเทียบกับ Ideal Timetable ที่มีการสร้างไว้เพื่อตรวจสอบ



รูปที่ 4.6 Simulate Timetable

4.3 ระบบจำลองการเดินรถ

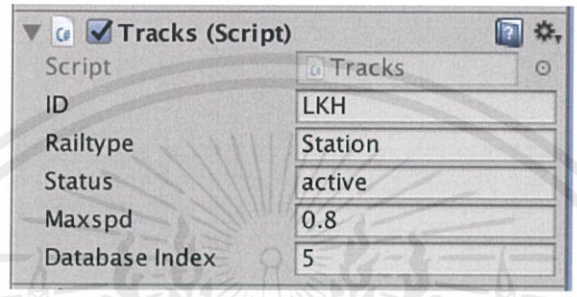
เป็นส่วนหนึ่งของระบบอาณัติสัญญาณซึ่งมีการนำมาใช้งานในระบบจำลองซึ่งเป็นองค์ประกอบหลักใหญ่ๆ ของระบบจำลองโดย ทั้งสองโมดูลนี้จะทำหน้าที่ในการจองราง กำหนดทิศทาง และเส้นทางเดินรถ รวมไปถึงสั่งหยุดรถ หรือสับเปลี่ยนเส้นทาง

4.3.1 Track Module

เป็นองค์ประกอบหลักของระบบจำลองโดยในตัว Track จะมีการเก็บค่าต่างๆ เพื่อนำไปประมวลผล เพื่อให้โมดูลรถไฟนำไปใช้งานโดยองค์ประกอบหลักของ Track จะมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Name
- ID
- Railtype
- Status
- Maxspd
- Database Index (Optional)

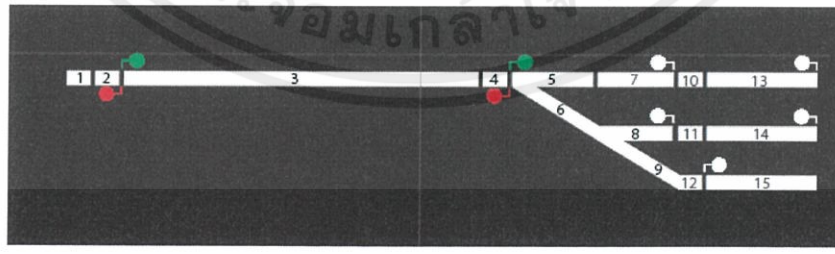


รูปที่ 4.7 การตั้งค่าการทำงานของ Track (ในโปรแกรม Unity)

ซึ่งค่าในแต่ละแอททริบิวนั้นจะไม่แสดงผลในตัวหน้าต่างแสดงผลของผู้ใช้ ซึ่งหากจำเป็นต้องทำการปรับค่า หรือสร้างส่วนต่อขยายเพิ่มนั้น สามารถแก้ไขได้ในโปรแกรม Unity โดยแต่ละแอททริบิวของ Track นั้นสามารถอธิบายได้ดังนี้

4.3.1.1 Track Name

ชื่อของวัตถุใน Unity ซึ่งเป็นค่าที่ใช้ในการระบุเป้าหมายของรถไฟ ซึ่งในการสร้าง Track ของโปรแกรมนั้นจะต้องสร้างโดยทำการไล่ลำดับจากซ้ายไปขวาบนลงล่างซึ่งในTrain Module จะมีการตรวจสอบรางทุกตัวก่อนจะทำการวิ่งเสมอโดยจะทำการตรวจสอบรางที่ถูกจองจากชื่อของรางและสถานะของรางและนำไปประมวลผลก่อนจะทำการวิ่งตามเส้นทางที่กำหนดไว้



รูปที่ 4.8 Track Naming

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

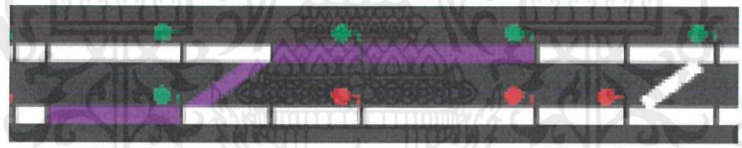
4.3.1.2 Track ID

เป็นค่า ID ของ Track ที่มีการใช้งานในส่วนของ Station เพื่อระบุว่า Station นี้ มี ID อะไร ซึ่งจะนำไปใช้ในการจับเวลา ว่ารถไฟมีการเข้าถึงสถานี ณ เวลาเท่าใด และทำการบันทึกลงใน Simulate Timetable เพื่อนำไปประเมินผล

4.3.1.3 Track Type (Railtype)

Rail Type แบ่งได้เป็น 3 ประเภทใหญ่ๆ

- **Track** เป็น Track Module ปกติทั่วไป
- **Switch** เป็น Track Module ที่ใช้ในการสับราง โดย Track Module จะไม่มีการเชื่อมกันหากยังไม่ทำการเลือกเส้นทาง ซึ่งเมื่อทำการจองเส้นทางแล้ว Track Module จะทำการสับให้โดยอัตโนมัติ โดยจากภาพจะเห็นได้ว่า Track Module ทางด้านซ้ายมีการจองรางด้วยรถไฟ แต่ทางด้านขวาไม่มีการจอง ทำให้รางสับทางด้านขวา ไม่มีการเชื่อมกับรางหลัก
- **Station** เป็น Track Module ที่เป็น Platform ของรถไฟ ซึ่งเป็นจุดที่รถไฟจะจอดที่สถานี ตามระยะเวลาที่กำหนดเอาไว้ และเป็น Track Module ที่ใช้เก็บค่าเวลาที่รถเข้าถึงสถานีด้วย เช่นกัน



รูปที่ 4.9 Track Switching

4.3.1.4 Track Status

เป็นค่าแอททริบิวสำคัญของระบบ โดยจะใช้ในการจองราง การตรวจสอบเส้นทางของรถ โดยในส่วนของ Track Status จะแบ่งได้คร่าวๆ ดังนี้

- **Active** เป็นค่าเริ่มต้นของ Track ทั้งหมด โดยจะมีค่าเป็นรางว่าง และยังไม่ถูกจองโดยรถไฟใดๆ ซึ่งจะสามารถใช้ Track Reservation Module ในการจองรางได้ ซึ่งสีของ Track ที่เป็น Active จะเป็นสีขาว
- **reserveX** เป็นค่าสถานะของรางที่ถูกจองโดยรถไฟ โดย X จะเป็นค่าตัวแปรที่เป็นชื่อรถไฟ ซึ่งเปลี่ยนไปตามรถไฟที่ใช้ โดยในที่นี้ มีรถไฟในโปรแกรม 3 คัน คือ A, B, C (TX1,TX2,TX3 ตามลำดับ) ดังนี้ค่า reserveX ในโปรแกรมจะมีค่าเป็น reserveA, reserveB, reserveC และมีสีรางคือ ฟ้า,เหลือง และม่วง ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Broken** เป็นค่าสถานะที่รางไม่สามารถใช้งานได้ ซึ่งมักจะเกิดจากเหตุการณ์ที่ตั้งไว้ โดยในกรณีที่รางเป็นสถานะ Broken เราจะไม่สามารถควบคุม Track นั้นๆได้ และจำเป็นต้องบังคับให้รถไฟวิ่งไปใช้เส้นทางอื่นแทน เพื่อป้องกันอุบัติเหตุที่จะเกิดขึ้น โดยสีของรางเมื่อพังจะเป็นสีเทา

ค่าสถานะของรางสามารถเปลี่ยนได้หลายวิธี

1.) **Manual** สามารถทำได้โดยการคลิก ที่ Track โดยจะ เปลี่ยนค่าสถานะตามลำดับดังนี้Active > reserveA > reserveB > reserveC > Active

2.) **Track Reservation Module** สามารถทำได้โดยใช้การเลือกรถไฟที่ต้องการ , สถานีต้นทาง และสถานีปลายทาง โดยระบบจะทำการตรวจสอบว่ามีเส้นทางมีการจองอยู่หรือไม่ ถ้าเกิดสามารถจองได้ จะทำการจองเส้นทางที่เป็นค่าเริ่มต้นของแต่ละสายให้โดยอัตโนมัติ

3.) **Train Module** โดยปกติแล้ว Train Module จะวิ่งไปตามเส้นทางที่ถูกจองเอาไว้ ซึ่งเมื่อรถไฟวิ่งผ่านเส้นทางที่ถูกจองไปแล้ว จะมีการคืนเส้นทางโดยอัตโนมัติ ซึ่งจะทำการคืนค่าของสถานะจาก reserveX เป็น Active เสมอ

4.3.1.5 Track Max Speed (MaxSpd)

เป็นค่า Max speed ของ Track ที่สามารถกำหนดได้ โดยจะกำหนดตามประเภทของ Track และระยะทางของ Track นั้นๆ ซึ่งจะมีค่าต่างกันไป เช่นในส่วนของทางโค้ง หรือเส้นทางช่วงสับ Track จะมีค่า maxspeed ที่ต่ำ เพื่อป้องกันอุบัติเหตุต่างๆ แต่ใน Track ที่เป็นทางตรงจะมีค่า Maxspd ที่สูง

4.3.1.6 Database Index

เป็นการดึงค่าแอททริบิวของ Track ข้างต้นจาก Database ซึ่ง จะต้องทำการดึงค่ามาโดยใช้ php file ในการดึงข้อมูลจาก Database มาเป็น String แล้วนำค่าที่ได้ ไปเก็บไว้ใน ซึ่งอาเรย์ Database Index ที่จะนำมาใส่ จะเป็นอาเรย์ Index ของ Track นั้นๆ ที่ถูกเก็บไว้ในอาเรย์ ซึ่งสามารถเลือกใช้วิธีนี้ หรือจะทำการป้อนค่าโดยตรงก็ได้เช่นกัน

4.3.2 Train Module

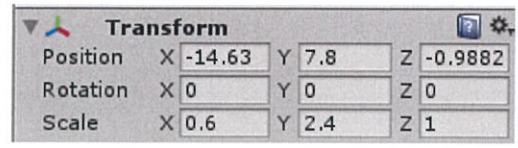
เมื่อทำการกดเริ่มการจำลองตัว Train Module จะเป็นโมดูลจำลองการวิ่งของรถไฟ ซึ่งในโปรแกรมของเรานั้นเหตุการณ์ต่างที่เกิดขึ้นจะมี Train Module เป็นโมดูลหลักที่เกี่ยวข้องทั้งสิ้น เพราะฉะนั้นสคริปการทำงานหลักจะอยู่ใน Train Module เกือบทั้งหมด

Train Module จะมีส่วนประกอบของโมดูลดังนี้

4.3.2.1 Transform

เป็นตัวระบุตำแหน่งและขนาดของ Train Module ในโปรแกรม

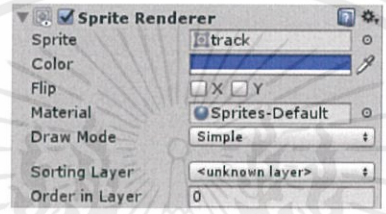
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 Transform Component (in Unity)

4.3.2.2 Sprite Renderer

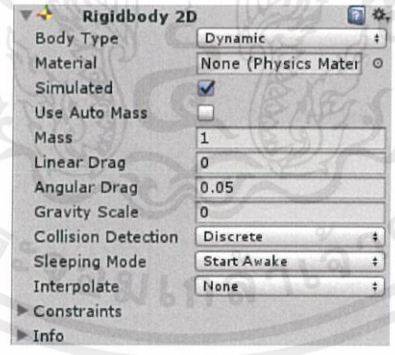
เป็นตัวกำหนดลักษณะการแสดงผล รูปร่าง และสีของ Train Module



รูปที่ 4.11 Sprite Renderer Component (in Unity)

4.3.2.3 Rigidbody 2D

เป็นส่วนประกอบที่ทำหน้าที่เกี่ยวกับการเกิด Collision

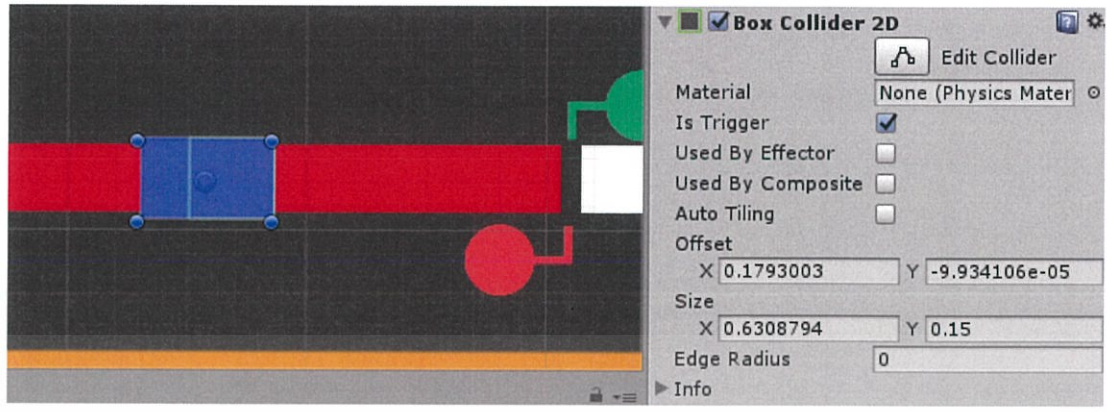


รูปที่ 4.12 Rigidbody Component (in Unity)

4.3.2.4 Collider 2D

เป็นตัวตรวจสอบการเกิด Collision ซึ่งจะมีขนาดเท่าๆกับ Train Module

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 Collider Component (in Unity)

4.3.2.5 Train Script

เป็นสคริปที่ทำให้โมดูลรถไฟสามารถทำงานได้โดยใช้ส่วนประกอบของโมดูลรถไฟเองและส่วนประกอบของโมดูลอื่นในโปรแกรมมาใช้ในการทำงานด้วย

รายละเอียดฟังก์ชันใน Train Script

4.3.2.5.1 ส่วนที่เกิดการทำงานอัตโนมัติเมื่อเริ่มใช้โมดูลรถไฟ (void Start())

จะมีการเก็บข้อมูลเวลาจากโมดูลนาฬิกาเพื่อนำมาใช้ในการระบุเวลาเมื่อรถไฟเข้าจอดสถานี

4.3.2.5.2 ส่วนที่มีการทำงานตลอดเวลาเมื่อโมดูลรถไฟเปิดใช้งาน (void FixedUpdate())

ส่วนนี้จะใช้ฟังก์ชันที่ต้องมีการทำงานตลอดเวลาเป็นหลัก ซึ่งกรณีนี้จะใช้ในการตรวจสอบสัญญาณไฟที่อยู่ในระยะ การวิ่งของรถไฟ และการอัพเดตข้อมูลการจอง Track ในขณะนั้นๆ

4.3.2.5.3 ส่วนที่มีการเช็ค Coliision (OnTriggerEnter(), OnTriggerExit()) จาก Collider ของโมดูลรถไปกับ Collider ของโมดูลอื่นๆในโปรแกรม

ฟังก์ชันนี้จะทำงานเมื่อ Collider ของโมดูลรถไฟกับ Collider ของโมดูลอื่นๆในโปรแกรมสัมผัสกันในสคริปนี้จะใช้ในการเช็คสัญญาณไฟเพื่อให้รถไฟหยุดหรือวิ่งตามสัญญาณที่เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2.5.4 stopforsec()

เป็นฟังก์ชันหยุดการเดินของรถไฟในกรณีที่ต้องหยุดรถตามเวลาที่กำหนด ซึ่งกรณีนี้ใช้กับการจอดของรถไฟเมื่อถึงสถานี ในฟังก์ชันจะมีการดึงข้อมูลเวลามาใช้ในการแสดงผลเวลาเมื่อรถไฟถึงสถานีนั้นๆด้วย

4.3.2.5.5 ConvertTime()

เป็นฟังก์ชันแปลงค่าเวลาในโปรแกรมให้ออกมาเป็นหน่วยชั่วโมง นาที และวินาที

4.3.2.5.6 Checkstation()

เป็นฟังก์ชันที่ใช้เช็คว่ารางต่อไปเป็นรางที่อยู่ในสถานีรึเปล่า ถ้าใช่จะให้มีการหยุดจอดโดยนำฟังก์ชัน stopforsec() มาใช้ในการหยุดรถไฟ

4.3.2.5.7 CheckSignal()

เป็นฟังก์ชันที่ใช้ในการนำสัญญาณไฟที่รถไฟได้รับจากการ Collision มาตรวจสอบว่าเป็นสัญญาณรถหยุดหรือไม่ ถ้าใช่ก็ต้องให้รถไฟหยุดจนกว่าสัญญาณไฟจะเปลี่ยนให้วิ่งต่อได้ ซึ่งในการที่จะให้ฟังก์ชันนี้ตรวจสอบได้ตลอดเวลาจึงต้องนำไปใช้ในส่วนที่มีการทำงานตลอดเวลาเมื่อโมดูลรถไฟเปิดใช้งาน (void FixedUpdate())

4.3.2.5.8 reloadRoute()

เป็นฟังก์ชันที่จะเข้าไปตรวจสอบข้อมูล Track ในโปรแกรมทั้งหมดเพื่อเก็บข้อมูลการจอง Track ที่ต้องการออกมาเป็น List<string> เช่น โมดูลรถไฟ A จะต้องใช้ฟังก์ชันนี้เพื่อเก็บข้อมูลการจอง Track เพื่อใช้ในการวิ่งของโมดูลรถไฟ จึงต้องหา Track ที่มีข้อมูลการจอง Track เป็น “reserveA” และฟังก์ชันนี้ต้องการให้มีการอัพเดตข้อมูลการจอง Track ตลอดเวลา เพราะรถไฟเมื่อวิ่งผ่าน Track ที่จองแล้วจะต้องปลดสถานะของการจองรางนั้นทันที ทำให้ฟังก์ชันนี้ต้องนำไปใช้ในส่วนที่มีการทำงานตลอดเวลาเมื่อโมดูลรถไฟเปิดใช้งาน (void FixedUpdate())

4.3.2.5.9 ChangeTarget()

เป็นฟังก์ชันที่นำข้อมูลการจอง Track ที่เก็บได้มาหาเป้าหมาย Track ถัดไปที่โมดูลรถไฟจะต้องวิ่งไป

4.3.3 Signal Module

เป็นส่วนหนึ่งของระบบอาณัติสัญญาณรถไฟ โดยใช้ในการควบคุม บอกเส้นทาง และหยุดรถไฟ โดย Signal จะเป็นส่วนที่ใช้ในการควบคุมสัญญาณเพื่อ บอกให้รถไฟเดินรถ หรือหยุด ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานของ Signal จะมีการควบคุม Headway ของ Signal เพื่อป้องกันไม่ให้เกิดการชนกันของรถไฟ ซึ่งจะกล่าวถึงในบทต่อๆไป

4.3.3.1 Signal Type

สามารถแบ่งได้เป็นสามประเภทใหญ่ๆ ที่นำมาใช้ในระบบจำลองการเดินรถ

1.) Two-way Signal เป็น Signal ที่ใช้ควบคุมรถรางเดี่ยว ซึ่งจะสามารถตรวจจับรถไฟได้จากทั้งสองทาง ซึ่งจะมีใช้งานในช่วงที่เป็นรางเดี่ยว ช่วง สถานีคลองตัน ถึงสถานีหัวหมาก

2.) One-way (To Right) Signal เป็น Signal ที่ใช้ควบคุมในส่วนของรางคู่ โดยจะบังคับให้เส้นทางนี้วิ่งได้แค่ทางเดียว จากซ้ายไปขวาเท่านั้น (จากตะวันตกไปตะวันออก) มีใช้งานในช่วงสถานีหัวหมาก ถึงสถานีบ้านทับช้าง

3.) One-way (To Left) Signal เป็น Signal ที่ใช้ควบคุมในส่วนของรางคู่ โดยจะบังคับให้เส้นทางนี้วิ่งได้แค่ทางเดียว จากขวาไปซ้ายเท่านั้น(จากตะวันออกไปตะวันตก) มีใช้งานในช่วงสถานีหัวหมาก ถึงสถานีบ้านทับช้าง

รูปที่ 4.14 Two-way Signal / To Right Signal/To Left Signal

4.3.3.2 Signal Attribute

ในการสร้าง Signal 1 ตัว จะประกอบไปด้วยแอททริบิวสองค่า คือสถานะ (status) และประเภท (Type) โดย Type จะเรียงตามลำดับจาก 4.3.3.1

- **Two-way signal**
- **One-way (To Right) Signal**
- **One-way (To Left) Signal**

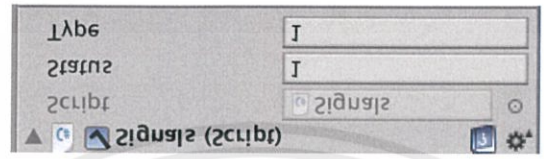
โดยค่าสถานะจะมีค่าต่างกันไปในแต่ละ Type ของ Signal ซึ่งสามารถเขียนได้เป็น Case ดังนี้

- **Type 1 , Status 0 :** รถไฟจากทางซ้ายหยุด รถไฟจากทางขวาผ่าน
- **Type 1 , Status 1 :** รถไฟจากทางซ้ายผ่าน รถไฟจากทางขวาหยุด
- **Type 2 , Status 0 :** รถไฟจากทางซ้ายหยุด
- **Type 2 , Status 1 :** รถไฟจากทางซ้ายผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Type 3 , Status 0 :** รถไฟจากทางขวาผ่าน
- **Type 3 , Status 1 :** รถไฟจากทางขวาหยุด

โดยมีข้อควรระวังในการสร้างคือ ใน Type 2,3 ค่าของ Status จะมีค่าตรงข้ามกันเสมอตามตารางด้านบน



รูปที่ 4.15 Signal Attribute (ในโปรแกรม Unity)

4.3.3.3 Headway & Signal Checking

ใน Signal 1 ตัว จะประกอบไปด้วย Collider ทั้งหมด 4 ตัว ซึ่งใช้ในการเช็คตำแหน่งของรถไฟ โดยจะประกอบไปด้วย

- **Left Collider** ด้านซ้ายสุด (ขีดสีเขียว) ติดกับ Signal ทางด้านซ้าย ใช้ตรวจจับว่ารถเข้ามายังเขตของเสาสัญญาณนี้แล้วหรือไม่ และตรวจจับว่า รถวิ่งพ้นเขตสัญญาณนี้แล้วหรือไม่เช่นกัน
- **Right Collider** ด้านขวาสุด (วงกลมสีเขียว) ติดกับ Signal ทางด้านขวา ใช้ตรวจจับว่ารถเข้ามายังเขตของเสาสัญญาณนี้แล้วหรือไม่ และตรวจจับว่า รถวิ่งพ้นเขตสัญญาณนี้แล้วหรือไม่เช่นกัน
- **Left Line of Sight Collider** อยู่ติดกับตัว Signal ทางด้านซ้าย (สี่เหลียมผืนผ้า) ใช้เป็นระยะมองเห็นของรถไฟ โดยเมื่อรถไฟเข้าสู่ระยะนี้แล้วหากสัญญาณไฟเป็นสีแดง จะทำการเบรคเพื่อหยุดรถไฟ
- **Right Line of Sight Collider** อยู่ติดกับตัว Signal ทางด้านขวา (สี่เหลี่ยมผืนผ้ากลม) ใช้เป็นระยะมองเห็นของรถไฟ โดยเมื่อรถไฟเข้าสู่ระยะนี้แล้วหากสัญญาณไฟเป็นสีแดง จะทำการเบรคเพื่อหยุดรถไฟ โดย Collider ทั้งสี่ตัวจะทำงานร่วมกันกับ Signal Status เพื่อตรวจสอบว่ารถไฟอยู่ที่จุดใดของช่วงสัญญาณ

โดยเมื่อรถไฟเข้าสู่เขต Signal ใหม่ (ผ่าน Left Collider) จะทำการปรับ Signal Status ของ Signal ที่อยู่ด้านหลังรถไฟให้เป็นสีแดงเสมอ เพื่อเป็นระยะ Headway ของรถไฟคันนั้นๆ และเมื่อรถไฟวิ่งไปถึงจุดสิ้นสุดของสัญญาณ (ผ่าน Right Collider) ซึ่งเป็นจุดเปลี่ยนเข้าสู่เขตสัญญาณใหม่ ก็จะทำการเปลี่ยน Status ของ Headway คืนกลับ ให้เป็นสีเขียวดังเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.16 Headway collider for Signal checking (in Unity)

4.3.4 Track Reservation Module

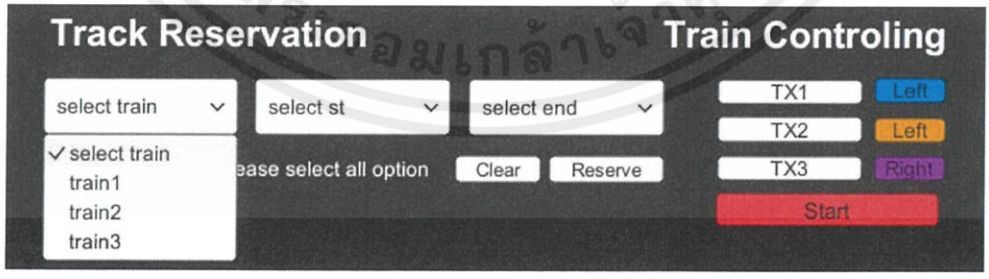
ระบบการจองรางมีหน้าที่ในการช่วยสำรองรางให้กับรถไฟที่จะทำการเดินรถโดยที่ผู้ใช้ไม่ต้องเลือกรางรถไฟที่ต้องการให้รถวิ่งที่ละรางด้วยตัวเอง โดยmoduleจะแบ่งออกเป็น 2 ส่วนหลักๆคือ การจองรางและการเคลียการจองรางทั้งหมด

4.3.4.1 Reserve track module

ใน module การจองรางนี้โปรแกรมจะรับตัวแปล input จากผู้ใช้ 3 ตัวแปล คือ

- Origin station
- Destination station
- Train

ผ่านทาง dropdown input ทั้ง 2 บนหน้าต่างแสดงผลของผู้ใช้จากนั้นโปรแกรมได้รับ input จากผู้ใช้ต่อเมื่อผู้ใช้กดปุ่มยืนยันการจองราง (Reserve) แล้วจากนั้นตัวโปรแกรมจะทำการเปลี่ยนสถานะของราง ทั้งหมดที่อยู่ในเส้นทางการเดินรถ ของแต่ละต้นทางปลายทางที่ ผู้ใช้ได้เลือกไว้ผ่านทางหน้าต่าง user interface โดยต้นทางคือค่าที่ได้รับจาก input Origin station ปลายทาง คือค่าที่ได้รับจาก Destination station จากนั้นโปรแกรมจะเปลี่ยนสถานะของรางให้ตรงตามชุดรถไฟที่ผู้ใช้เลือกไว้แล้วโมดูลรางรถไฟของโปรแกรม (track module) จะทำการแสดงผลการเปลี่ยนสถานะของรางต่อ (เปลี่ยนสีของรางตามสถานะของรางโดยอัตโนมัติ) โดยที่โปรแกรมจะรับค่าก็ต่อเมื่อ input ทั้งหมดถูกเลือกโดยผู้ใช้แล้วเท่านั้นและสถานีต้นทางกับปลายทางจะต้องไม่ซ้ำกันเท่านั้น



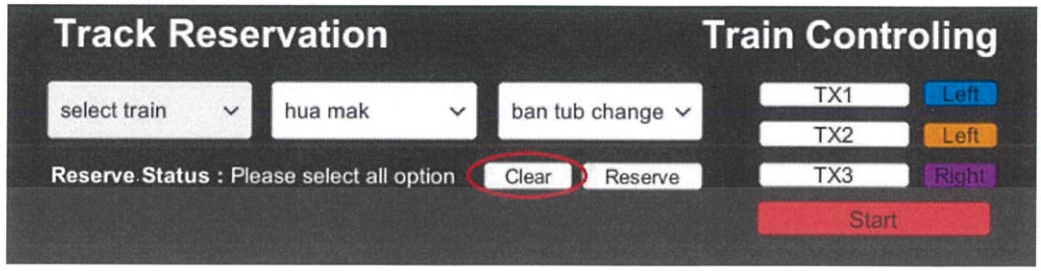
รูปที่ 4.17 Reserve track module

4.3.4.2 Clear track module

ในการที่จะลบค่าการจองรางทั้งหมดที่มีอยู่ผู้ใช้สามารถทำได้โดยการกดปุ่ม Clear บนหน้าต่าง user interface เมื่อปุ่มนี้มีการ active โปรแกรมจะทำการเปลี่ยนค่าสถานะของราง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

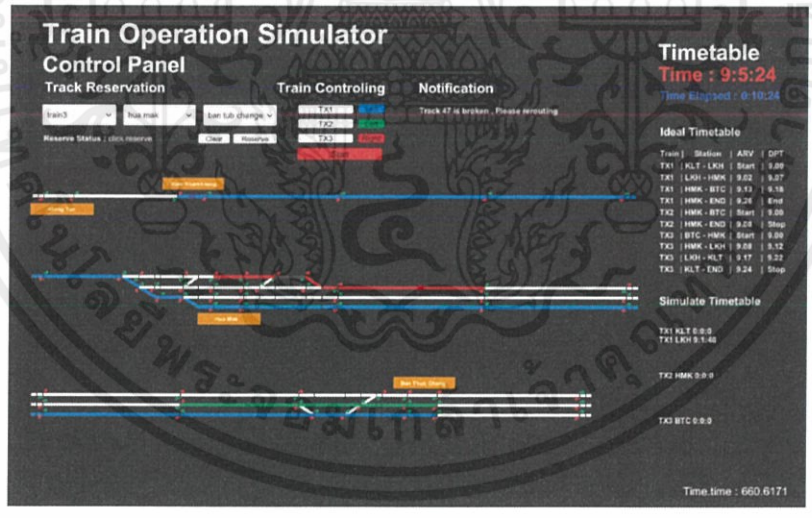
ทั้งหมดไม่ว่าตอนนั้นรางจะมีสถานะอะไรค้างอยู่ในฐานข้อมูลก็ตามให้เป็นว่าง (available) เพื่อช่วยให้ผู้ใช้สามารถทำการจองรางใหม่ได้ในกรณีที่มีการจองรางผิดพลาดหรือต้องการเปลี่ยนเส้นทาง



รูปที่ 4.18 Clear track module

4.3.5 Event Module

เป็นส่วนหนึ่งของระบบจำลองการเดินรถ ซึ่งโดยปกติแล้ว การควบคุมรถมักจะไม่มีเหตุผิดปกติร้ายแรงเกิดขึ้นบ่อยนัก แต่เพื่อสร้างทักษะและการตัดสินใจของผู้ควบคุมรถในระบบจำลองการเดินรถ เราจึงได้สร้าง Event มารองรับ เพื่อที่จะทดสอบการตัดสินใจ และการทำการแก้ไขปัญหาในที่จะเกิดขึ้นในระบบการควบคุมรถไฟ โดยเมื่อเกิด Event ขึ้น จะมีการแจ้งเตือนขึ้นมาในส่วนของ Notification Center



รูปที่ 4.19 Event module

4.3.5.1 Notification Center

เป็นส่วนที่ใช้ในการแจ้งเตือน เมื่อเกิดเหตุการณ์ต่างๆขึ้น โดยจำทำการเตือนว่าเกิดปัญหาที่จุดใด และรายละเอียดของสถานการณ์ที่เกิดขึ้น ซึ่งในจุดนี้เราอาจจะแก้ไขให้เพิ่มแนวทางการแก้ไขปัญหาเบื้องต้นเข้าไปได้หากต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5.2 Simulator Event

เป็นส่วนที่จะแสดงผลของ Event ที่จะเกิดขึ้นในระบบการควบคุมรถ ยกตัวอย่าง เช่น Event รางเสีย เราจะแทนรางที่เสียด้วยสีเทาดังภาพ 4.3.5 ซึ่งรางที่เสียนั้น จะไม่สามารถควบคุมได้ ซึ่งจำเป็นจะต้องสลับเส้นทางไปยังเส้นทางสำรอง หรือในกรณีที่แย่ที่สุดคือหยุดรถ เพื่อป้องกันอุบัติเหตุที่จะเกิดขึ้น และหากรถไฟเคลื่อนที่ไปยังรางที่เสีย จะเกิดอุบัติเหตุขึ้นและมีการแจ้งเตือนว่าผู้ทดสอบไม่ผ่านการทดสอบ

4.3.5.3 การสร้าง Event

ในการพัฒนาโปรแกรม ทางผู้พัฒนาได้ทำการเขียนสคริป Event.cs เพื่อใช้ในการสร้าง และใช้งาน Event ในตัวระบบจำลองโดย ใน Event.cs จะมี Script ในการจับเวลา เพื่อหาเวลาในการเริ่ม Event และ Parameter ของ Event ที่กำหนดว่าจะเกิดอะไรขึ้น ซึ่งสามารถจะปรับแก้ไขได้ใน Unity โดยเราสามารถกำหนด Parameter ต่างๆของระบบจำลองได้ โดยในการตั้งเวลาที่จะเกิดใน Event เราสามารถจะกำหนดเวลาให้เกิดตายตัว เมื่อเวลาผ่านไป x นาที โดยการกำหนดค่าโดยตรง หรือจะใช้ฟังชั่น Random() ที่เป็นฟังชั่นพื้นฐานของ C# เพื่อสุ่มเวลาใน range ที่เราต้องการออกมา เพื่อความหลากหลายของรูปแบบของ Event โดยใน Event เราสามารถตั้งได้ตั้งแต่ Track มีเสีย หรือรางสับมีปัญหา , Signal ไม่สามารถใช้งานได้และ รถไฟออกจากสถานีช้า เนื่องจากคนใช้งานเยอะเกินไป เป็นต้น

```
public class Event : MonoBehaviour {
    public float times = Time.deltaTime;
    public Text notification;

    // Use this for initialization
    void Start () {
        notification = GetComponent<Text>();
    }

    // Update is called once per frame
    void Update () {
        times += Time.deltaTime;
        Events1();
    }

    void Events1 (){
        Tracks brokentrack = GameObject.Find("Rail47").GetComponent<Tracks>();
        if (times >= 600f) {
            brokentrack.Status = "broken";
            Debug.Log (brokentrack);
            notification.text += "Track 47 is broken , Please rerouting";
        }
    }
}
```

รูปที่ 4.20 Event Class

4.3.6 Timetable & Evaluate Module

เป็นส่วนที่ใช้ตรวจสอบและวัดผลการทำงานของผู้ทดสอบการเดินรถ โดยเราจะวัดผลโดยการจับเวลาที่รถไฟเข้าจอดที่สถานีโดยสมบูรณ์ ซึ่งโดยปกติแล้ว การคิดคะแนนจากการทดสอบการเดินรถในรูปแบบนี้มักจะทำได้ยากจากการตัดสินผลโดยคอมพิวเตอร์ เราจึงได้ใช้วิธีเก็บค่าสถิติการใช้งานของผู้ทดสอบ เพื่อนำไปวัดผลด้วยผู้ควบคุมการทดสอบอีกทอดหนึ่ง โดยในส่วนของ Timetable จะสามารถแบ่งได้เป็นสองส่วนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6.1 Ideal Timetable

เป็นส่วนของ Timetable ที่ได้มีการทดสอบจากผู้พัฒนาแล้ว ว่าถ้าหากไม่เกิดปัญหาใดๆในระบบ รถไฟจะสามารถใช้ตารางเดินรถนี้ในการเดินรถได้อย่างมีประสิทธิภาพ โดยตารางเดินรถนี้ เกิดจากการทดสอบซ้ำหลายๆรอบ เพื่อตรวจสอบความถูกต้อง และแม่นยำโดยผู้พัฒนา

4.3.6.2 Simulate Timetable

เป็นส่วนของ Timetable ที่ผู้ทดสอบการเดินรถ ได้ใช้งานระบบจำลองการเดินรถควบคุม โดยจะบันทึกค่าเมื่อรถไฟเข้าสู่ชานชาลาของสถานี โดยจะบันทึกเวลาเมื่อรถไฟจอดสนิทที่ชานชาลาเท่านั้น ซึ่งในตารางส่วนนี้ จะประกอบไปด้วย

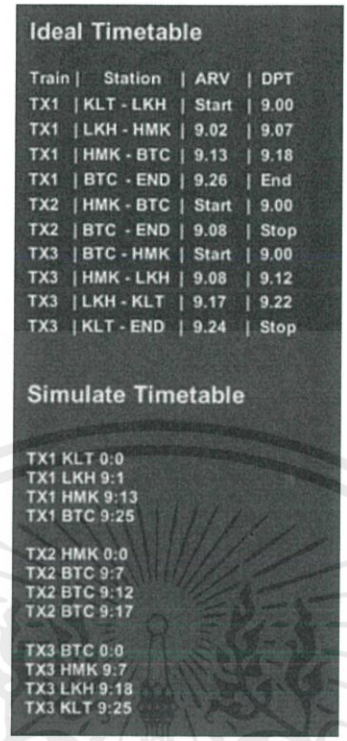
- Train ID
- Station ID
- Arrival Time

ตัวอย่างเช่น

TX1 BTC 9:25 คือรถไฟ TX1 ถึงสถานี BTC (บ้านทับช้าง) เมื่อเวลา 9.25

TX1 KLT 0:0 คือรถไฟ TX1 เริ่มที่สถานี KLT (คลองตัน) แต่เวลาจะแสดงเป็น 0 เนื่องจากเป็นสถานีเริ่มต้นของเส้นทางนั้นๆ ซึ่งจะเป็นกรณีพิเศษที่จะเกิดแค่ครั้งเดียวใน 1 เส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 Timetable Module

4.4 โปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถ

4.4.1 การทดลองระบบการจองรางด้วยต้นเอง

หลังจากที่ได้ทำการพัฒนาระบบจำลองในส่วนของระบบการจองรางแล้วได้ทำการทดสอบด้วยการเริ่มการจำลองแล้วคลิกที่รูปของรางรถไฟในหน้าต่างของระบบจำลองดูได้ผลออกมาว่าสีของรางรถไฟในหน้าต่างที่ได้ทำการคลิกนั้นมีค่าสถานะและสีที่เปลี่ยนไปตามที่ได้ตั้งเอาไว้

4.4.2 การทดลองระบบการจองรางด้วยโมดูล Track reservation

เมื่อเริ่มระบบจำลองการควบคุมและดูแลการเดินรถแล้วทำการเลือกชื่อสถานีต้นทางกับปลายทางและขบวนรถที่ต้องการในช่องรับอินพุดแบบดรอปดาวน์แล้วกดปุ่ม reserve แล้วจะเห็นว่าทุกรางที่อยู่ในเส้นทางระหว่างทั้งสองสถานีที่ได้ทำการเลือกไว้มีสีและค่าสถานะที่เปลี่ยนไปตามที่ต้องการ

4.4.3 การทดลองล้างค่าสถานะของการจองราง

เมื่อเริ่มการจำลองแล้วไม่ว่าจะมีรางใดๆที่อยู่ในระบบมีค่าสถานะเป็นอย่างไรก็ตามถ้าหากเรากดที่ปุ่ม clear แล้วทุกๆรางจะเปลี่ยนค่าสถานะเป็น Active ตามที่ได้เขียนคำสั่งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.4 การทดลองโมดูลรถไฟ

เมื่อเริ่มการจำลองหลังจากที่ได้ทำการจองรางตามเส้นทางต่างๆได้แล้วกดปุ่ม start จะเห็นว่ารถไฟสามารถวิ่งไปตามเส้นทางที่เราได้ทำการจองไว้ได้ตรงตามขบวนรถและตรงตามเส้นทางที่เราได้ทำการจองเอาไว้อย่างแม่นยำ

4.4.5 การทดลองจับเวลาด้วยโมดูลจับเวลา

หลังจากที่เราทดลองการวิ่งของโมดูลรถไฟแล้วสามสารถสังเกตตารางเวลาด้านขวาของหน้าต่างระบบจำลองได้ว่าโมดูลเวลาสามารถจับเวลาได้ว่ารถขบวนต่างๆเข้าถึงสถานีต่างๆที่เวลาได้บ้างอย่างแม่นยำ

4.4.6 การทดลองตรวจสอบการทำงานของระบบอาณัติสัญญาณจำลอง

หลังจากที่เราทดลองการวิ่งของโมดูลรถไฟแล้วสามสารถสังเกตได้ว่าเมื่อตัวรถนั้นวิ่งผ่านสัญญาณไฟที่อยู่ระหว่างเส้นทางการเดินรถตามจุดต่างๆของระบบจำลองแล้วสัญญาณไฟที่แสดงในหน้าต่างของระบบจำลองมีการเปลี่ยนแปลงถูกต้องตามที่เราได้ตั้งค่าเอาไว้ในระบบจำลองอย่างแม่นยำ

4.4.7 การทดลองสับรางรถไฟ

หลังจากที่เราทดลองการวิ่งของโมดูลรถไฟแล้วต้องการสับเปลี่ยนเส้นทางการวิ่งของรถไฟเราสามารถเปลี่ยนเส้นทางการวิ่งของรถไฟได้ตามที่เราต้องการภายใต้เงื่อนไขของระยะทางที่เราได้ทำการตั้งระบบตรวจจับตำแหน่งของขบวนรถว่ามีระยะที่เพียงพอต่อการเปลี่ยนเส้นทางการเดินรถไปยังรางอื่นหรือไม่

4.4.8 การทดลองโมดูลสถานการณ์จำลอง

จากการที่เราได้ตั้งให้มีเหตุการณ์รางเสียหายไว้ตามเวลาที่กำหนดในระบบจำลองนี้ทำให้รถไม่สามารถวิ่งเส้นทางที่ผ่านรางที่เสียหายนั้นได้เมื่อเราเริ่มทำการทดลองใช้ระบบจำลองดูสังเกตได้ว่าเมื่อถึงเวลาที่กำหนดรางที่ตั้งเอาไว้มีสถานะเปลี่ยนไปจริงๆตามที่เราตั้งใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงงานนี้นำเสนอในเรื่องของการพัฒนาโปรแกรมระบบจำลองการควบคุมและดูแลการเดินรถโดยมีการกำหนดให้โปรแกรมทำงานบนคอมพิวเตอร์ที่ทำการติดตั้งโปรแกรมแล้วโดยไม่ต้องการการเชื่อมต่อระบบอินเตอร์เน็ตเพื่อใช้งานโปรแกรมเพื่อให้ผู้ใช้เข้าถึงได้ง่ายแม้ว่าไม่มีระบบอินเตอร์เน็ตก็สามารถใช้งานโปรแกรมได้โดยตัวโปรแกรมนั้นสามารถจำลองสถานการณ์ต่างๆที่สามารถเกิดขึ้นได้ในการทำงานจริงของผู้ควบคุมระบบการเดินรถทั้งสภาวะปกติและสภาวะที่ไม่ปกติทำให้ผู้ใช้โปรแกรมได้ฝึกฝนความชำนาญในการปฏิบัติงานและยังประเมินผลการทำงานโดยรวมให้แก่ผู้ใช้งานโปรแกรมได้ในระดับหนึ่ง

5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

เนื่องจากโปรแกรมที่พัฒนาขึ้นมายังมีขีดจำกัดในหลายๆ ด้านเช่น

- การสุ่มของเหตุการณ์ที่ไม่ปกติยังสามารถเกิดได้เฉพาะจุดที่ตั้งไว้และยังเวลาเกิดได้แค่ช่วงที่เราตั้งไว้จึงทำให้ผู้ใช้อาจสามารถจดจำรูปแบบได้หากมีการใช้งานโปรแกรมบ่อยๆ
- ในเวอร์ชันปัจจุบันที่ได้ทำการพัฒนานั้นถูกออกแบบให้ผู้ใช้นั้นมีสองขนาดการแสดงผลเพื่อให้ตรงกับขนาดการแสดงผลของคอมพิวเตอร์ของผู้ใช้ยังถือเป็นข้อเสียที่ผู้ใช้ต้องเลือกใช้งานระบบจำลองที่มีการแสดงผลตรงกับคอมพิวเตอร์ของผู้ใช้เองเนื่องจากการออกแบบของผู้พัฒนาระบบจำลองจะไม่สามารถเปลี่ยนแปลงแก้ไขโดยไม่ส่งผลกระทบกับระบบจำลองได้
- การประเมินผลการควบคุมประสิทธิภาพในการควบคุมรถยังเป็นเพียงการเปรียบเทียบเวลาที่ผู้เรียนได้จากการจำลองการควบคุมรถ
- ยังไม่สามารถสร้างเส้นทางการเดินรถใหม่ๆ ได้แบบไดนามิคทำให้การเพิ่มเส้นทางจะต้องมีการวางโครงสร้างของเส้นทางการเดินรถใหม่ทำให้เกิดความยุ่งยากให้การขยายโปรแกรม

5.3 แนวทางพัฒนาต่อไป

โครงงานนี้สามารถพัฒนาต่อไปได้หลากหลายรูปแบบเช่น

- เพื่อเป็นการฝึกผู้เรียนให้ได้ดียิ่งขึ้นควรเพิ่มให้เป็นรูปแบบการสุ่มของเหตุการณ์ที่ผิดปกติทั้งตำแหน่งในการเกิดเหตุการณ์และเวลาในการเกิดเหตุการณ์เพื่อให้เกิดความแปลกใหม่ในการใช้โปรแกรมจำลองเช่นมีรูปแบบการสุ่มเวลาการเกิดเหตุการณ์ทำให้การประเมินผู้ใช้มีประสิทธิภาพมากขึ้นไปอีกเนื่องจากมีความแตกต่างในการทดลองใช้ระบบจำลองในแต่ละครั้ง
- เพิ่มเส้นทางการเดินรถจากเดิมที่มีอยู่แล้วให้มีเส้นทางที่แตกต่างมากขึ้นทำให้ผู้ใช้ได้ฝึกฝนการควบคุมการเดินรถในรูปแบบใหม่ๆ
- เพิ่มระบบการประเมินผู้ใช้ระบบจำลองจากที่เป็นเพียงเก็บบันทึกผลการจำลองในรูปแบบเวลาเป็นการนำมาคำนวณและประมวลผลให้เห็นได้อย่างชัดเจนเป็นระดับ A, B, C, D

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พัฒนาโครงสร้างของโปรแกรมให้มีการสร้างเส้นทางการเดินรถใหม่แบบไดนามิคจากเดิมที่เป็นแบบสแตติกเพื่อให้ขยายระบบได้ง่ายยิ่งขึ้น
- ในการพัฒนาต่อให้ระบบมีความไดนามิคมากขึ้นอาจจะต้องมีการวางโครงสร้างใหม่ให้มีลำดับการเรียงของหมายเลขรางรถไฟให้เรียงกันเป็นลำดับที่ถูกต้องเพื่อให้สามารถเขียนโมดูลการจองรางให้เป็นแบบไดนามิคได้เพื่อความง่ายยิ่งขึ้นในการขยายโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Matthew-Schell. 2014. 2D UFO Tutorial : https://unity3d.com/learn/tutorials/projects/2d-ufo-tutorial

วันวิสา ชัชวงษ์. 2599. การจัดสร้างห้องปฏิบัติการศูนย์วิจัยการเรียนรู้การบริหารและควบคุมระบบการเดินรถไฟ: http://bit.ly/2pswqUY

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

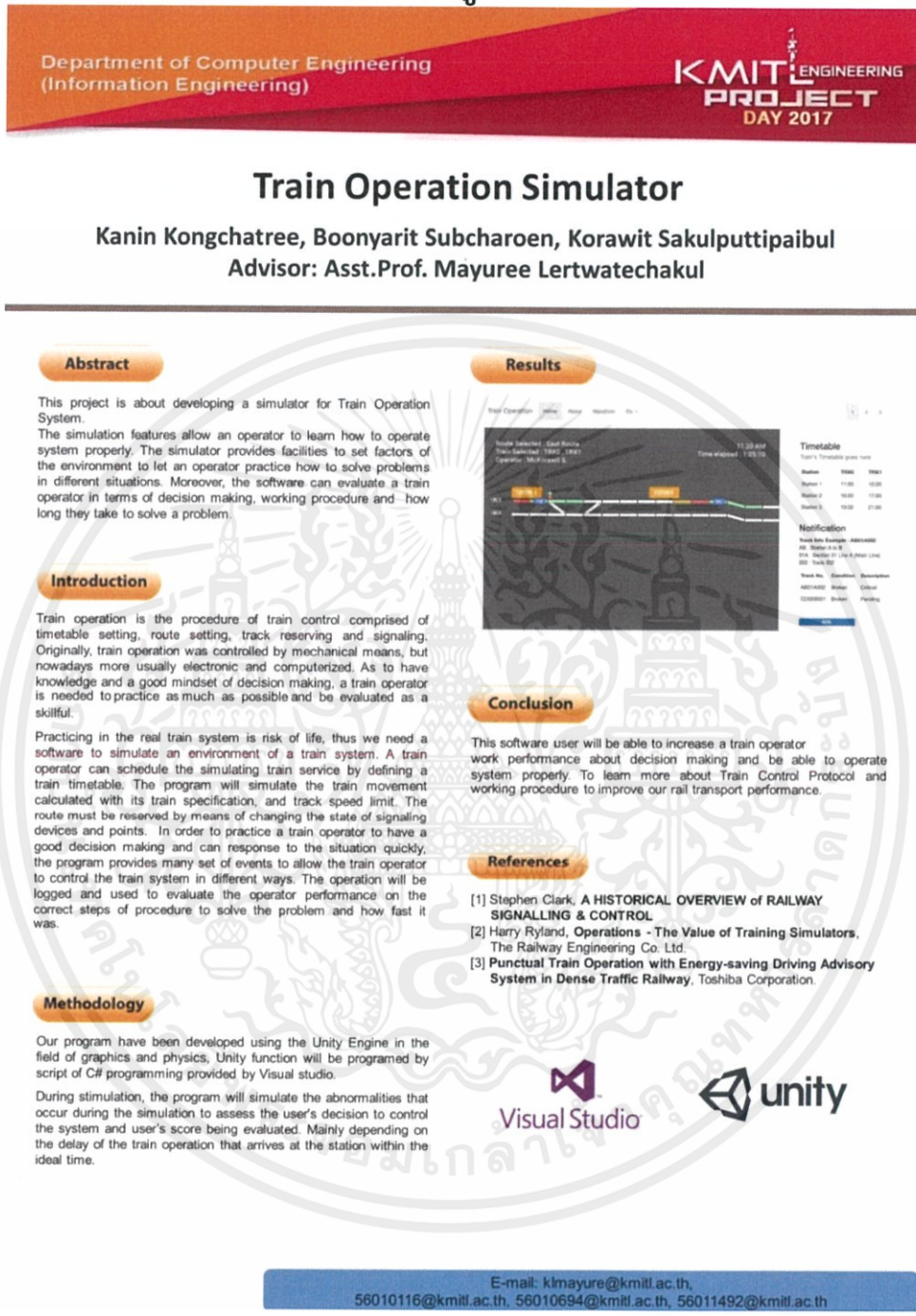
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
Poster (และรูปผลงานถ้ามี)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Poster (และรูปผลงานถ้ามี)



Department of Computer Engineering
(Information Engineering)

KMITL ENGINEERING PROJECT DAY 2017

Train Operation Simulator

Kanin Kongchatree, Boonyarit Subcharoen, Korawit Sakulputtipaibul
Advisor: Asst.Prof. Mayuree Lertwatechakul

Abstract

This project is about developing a simulator for Train Operation System.
The simulation features allow an operator to learn how to operate system properly. The simulator provides facilities to set factors of the environment to let an operator practice how to solve problems in different situations. Moreover, the software can evaluate a train operator in terms of decision making, working procedure and how long they take to solve a problem.

Introduction

Train operation is the procedure of train control comprised of timetable setting, route setting, track reserving and signaling. Originally, train operation was controlled by mechanical means, but nowadays more usually electronic and computerized. As to have knowledge and a good mindset of decision making, a train operator is needed to practice as much as possible and be evaluated as a skillful.

Practicing in the real train system is risk of life, thus we need a software to simulate an environment of a train system. A train operator can schedule the simulating train service by defining a train timetable. The program will simulate the train movement calculated with its train specification, and track speed limit. The route must be reserved by means of changing the state of signaling devices and points. In order to practice a train operator to have a good decision making and can response to the situation quickly, the program provides many set of events to allow the train operator to control the train system in different ways. The operation will be logged and used to evaluate the operator performance on the correct steps of procedure to solve the problem and how fast it was.

Methodology

Our program have been developed using the Unity Engine in the field of graphics and physics, Unity function will be programed by script of C# programming provided by Visual studio.

During stimulation, the program will simulate the abnormalities that occur during the simulation to assess the user's decision to control the system and user's score being evaluated. Mainly depending on the delay of the train operation that arrives at the station within the ideal time.

Results

Conclusion

This software user will be able to increase a train operator work performance about decision making and be able to operate system properly. To learn more about Train Control Protocol and working procedure to improve our rail transport performance.

References

[1] Stephen Clark, **A HISTORICAL OVERVIEW of RAILWAY SIGNALLING & CONTROL**
[2] Harry Ryland, **Operations - The Value of Training Simulators**, The Railway Engineering Co. Ltd.
[3] **Punctual Train Operation with Energy-saving Driving Advisory System in Dense Traffic Railway**, Toshiba Corporation.

Visual Studio unity

E-mail: klmayure@kmitl.ac.th,
56010116@kmitl.ac.th, 56010694@kmitl.ac.th, 56011492@kmitl.ac.th

รูปที่ ก.1 Poster Train Operation Simulator

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
การติดตั้งโปรแกรม Unity

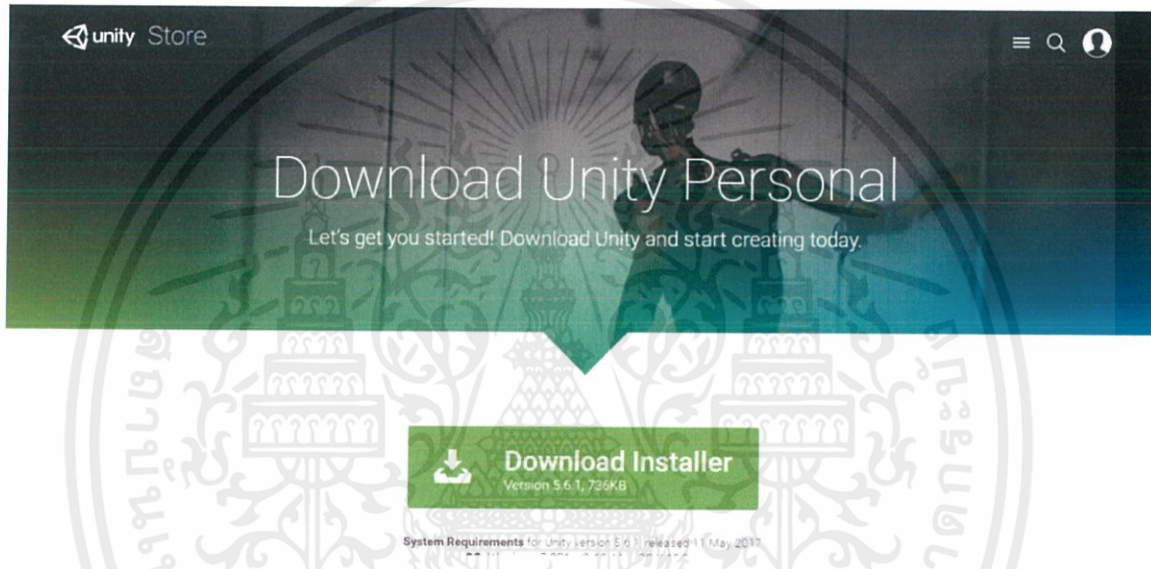
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
ตัวอย่าง การติดตั้งโปรแกรม Unity

ขั้นตอนการติดตั้งโปรแกรม Unity

โปรแกรมบนระบบปฏิบัติการ Windows สามารถอธิบายได้ดังนี้

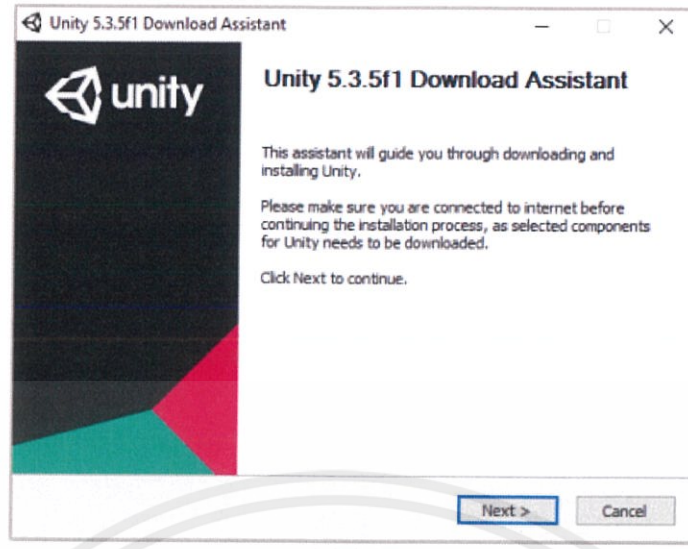
1. สามารถเข้าไปดาวน์โหลดตัวโปรแกรมได้จาก
https://store.unity.com/download?ref=personal ดังรูปที่ ข.1



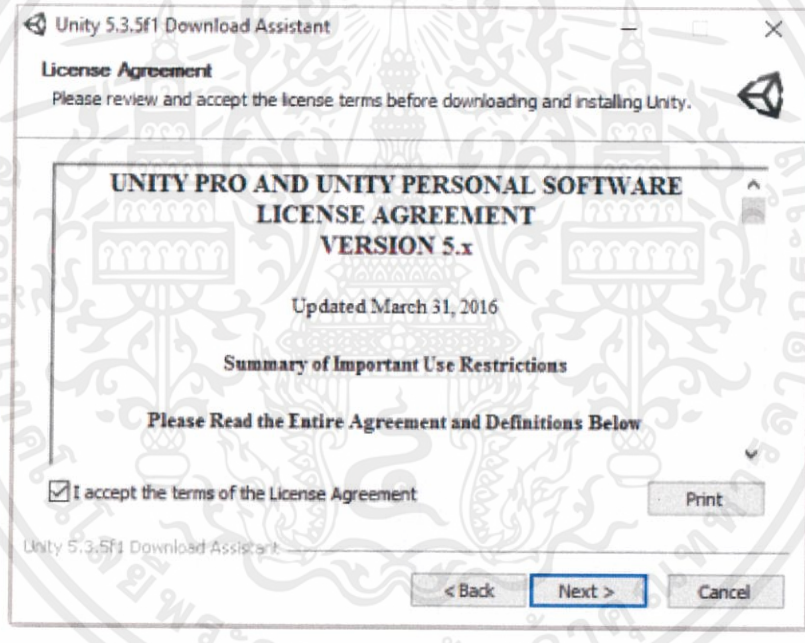
รูปที่ ข.1 หน้าเว็บไซต์แสดงขั้นตอนการเข้าไปดาวน์โหลดโปรแกรม Unity

2. เมื่อทำการดาวน์โหลด Unity มาเรียบร้อยแล้วเปิดตัวติดตั้งโปรแกรมขึ้นมากดยอมรับข้อตกลงในการเข้าใช้งานให้เรียบร้อยดังรูปที่ ข.2 และ ข.3

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 หน้าต่างเมื่อเปิดตัวติดตั้งโปรแกรม Unity ขึ้นมาแล้ว



รูปที่ ข.3 ตัวอย่างหน้าต่างยอมรับเงื่อนไขการใช้งานโปรแกรม Unity

3. เลือก Components ตรงนี้เป็นส่วนที่ควรจะให้ความสำคัญในตอนติดตั้ง เพราะเป็นส่วนที่เราจะให้เลือกว่าเราต้องการให้ติดตั้งอะไรเอาไว้ใช้งานบ้าง โดยจะมีการเลือกมาเป็นค่าเริ่มต้นมาให้ 5 อย่าง ได้แก่

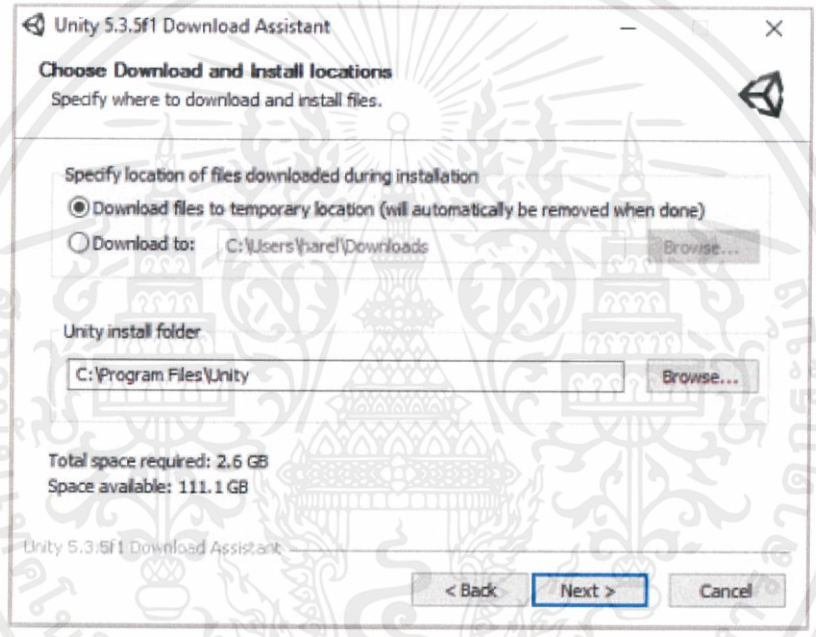
- Unity ซึ่งก็คือตัวโปรแกรม Unity เอง
- Documentation เอกสารประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Web player เป็นตัวเล่นโปรแกรมที่พัฒนาขึ้นจาก Unity ผ่านเว็บบราวเซอร์
- Standard Assets เป็นพวก Assets ที่ทาง Unity เตรียมไว้ให้ใช้งาน
- Windows Build Support เป็น Components ที่เอาไว้สำหรับรองรับการพัฒนาเกมบนระบบปฏิบัติการ Windows

โดยผู้ใช้สามารถเพิ่มส่วนเสริมที่ต้องการใช้งานเองได้ตามต้องการเช่นต้องการพัฒนาลงระบบปฏิบัติการแอนดรอยก็ให้เลือก Android build support ไปด้วย

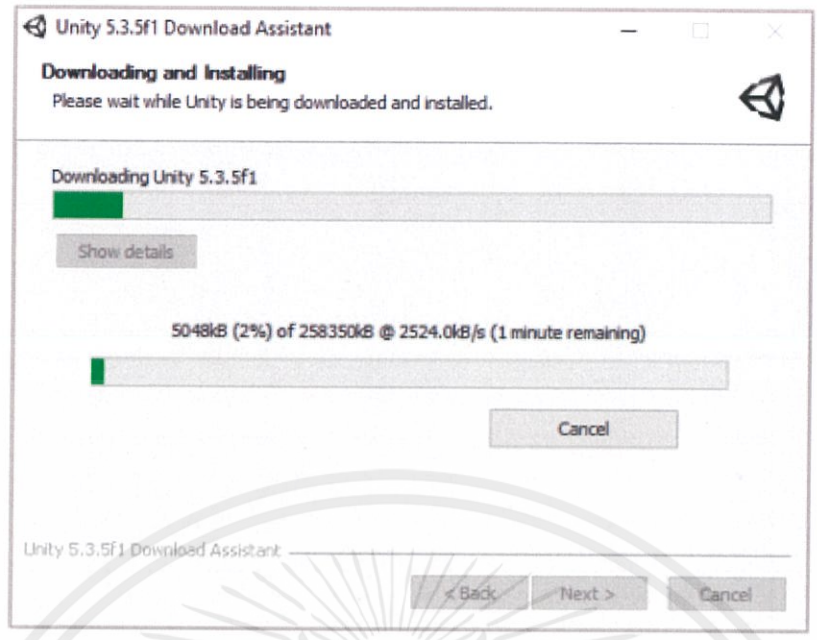
4.ทำการเลือกว่าเราจะทำการติดตั้งโปรแกรมแบบไหนโดยจะสามารถเลือกได้2แบบคือการดาวน์โหลดโปรแกรมแบบชั่วคราวและลบเมื่อทำการติดตั้งเสร็จแล้วหรือจะเลือกแบบดาวน์โหลดมาติดตั้งแล้วเมื่อติดตั้งเสร็จแล้วไฟล์ที่ใช้ในการติดตั้งก็จะยังคงอยู่ในเครื่อง



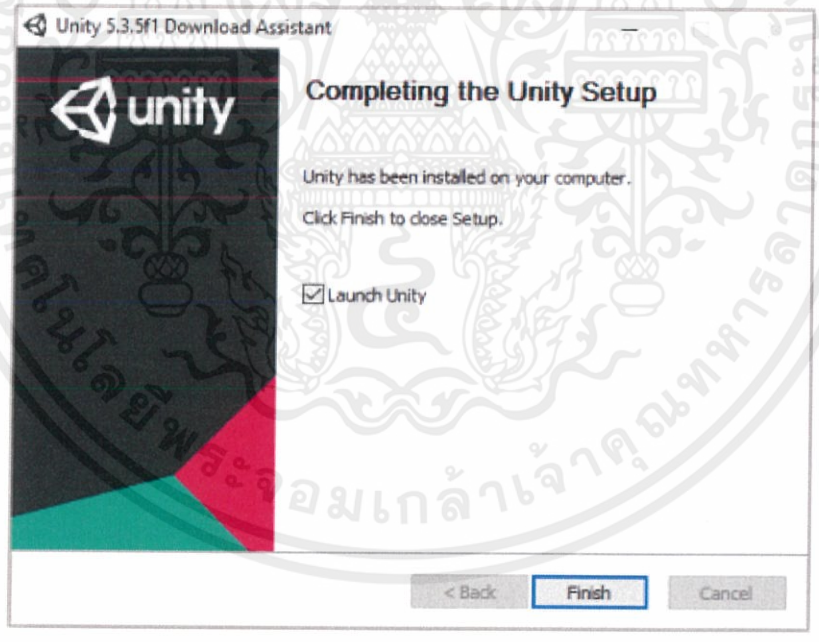
รูปที่ ข.4 หน้าต่างขั้นตอนเลือกประเภทในการติดตั้งโปรแกรม

5. รอจนโปรแกรมติดตั้งจนเสร็จก็เรียบร้อย พร้อมใช้งานได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ข.5 รูปตัวอย่างขณะทำการติดตั้งโปรแกรม



รูปที่ ข.6 รูปตัวอย่างเมื่อติดตั้งโปรแกรมเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้