การติดตั้งเครื่องมือเร็นเดอร์แผนที่บนระบบปฏิบัติการลีนุกซ์ และตั้งค่าให้รองรับภาพความละเอียดสูง

Set-up a Rendering Engine of MapQuest Open Locally And Configure it to serve HD tiles

นาย พงศกร อยู่สุข

รายงานฝึกงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยี ไทย-ญี่ปุ่น
พ.ศ. 2558

การติดตั้งเครื่องมือเร็นเดอร์แผนที่บนระบบปฏิบัติการลีนุกซ์และตั้งค่าให้รองรับภาพความละเอียดสูง

Set-up a Rendering Engine of MapQuest Open Locally and Configure it to serve HD tiles

นาย พงศกร อยู่สุข

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย - ญี่ปุ่น
ปีการศึกษา 2558

คณะกรรมการสอบ

..... ประธานกรรมการสอบ
(ดร.บุษราพร เหลืองมาลาวัฒน์)

..... กรรมการสอบ
(อาจารย์สลิลา ชีวกิดาการ)

..... อาจารย์ที่ปรึกษา
(อาจารย์อมรพันธ์ ชมกลิ่น)

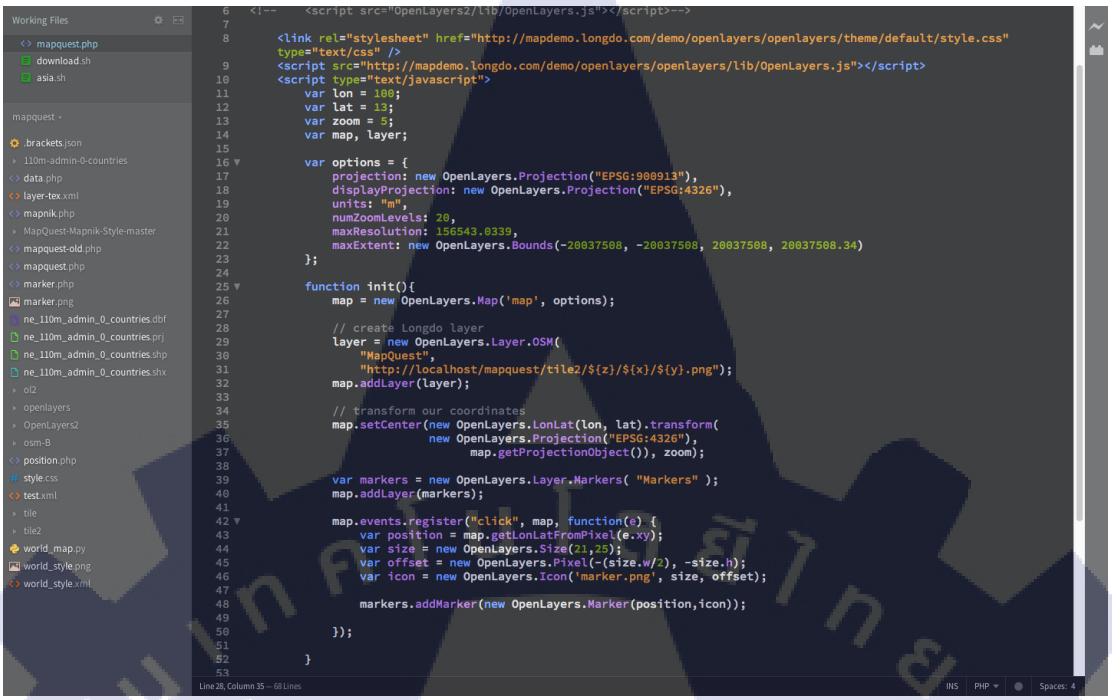
..... ประธานสหกิจศึกษาสาขาวิชา
(อาจารย์อมรพันธ์ ชมกลิ่น)

ชื่อโครงงาน/รายงาน	การติดตั้งเครื่องมือเร็นเดอร์แผนที่บนระบบปฏิบัติการลินุกซ์และตั้งค่าให้รองรับภาพความละเอียดสูง Set-up a Rendering Engine of MapQuest Open Locally and Configure it to serve HD tiles
ผู้เขียน	นาย พงศกร อยู่สุข
คณะวิชา	เทคโนโลยีสารสนเทศ สาขาวิชา เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษา	อาจารย์อมรพันธ์ ชมกลิ่น
พนักงานที่ปรึกษา	นาย ชุติภัทร์ โชคพิพัฒน์พร
ชื่อบริษัท	บริษัท เมตามีเดีย เทคโนโลยี จำกัด
ประเภทธุรกิจ/สินค้า	ให้การบริการด้านงานพัฒนาซอฟต์แวร์

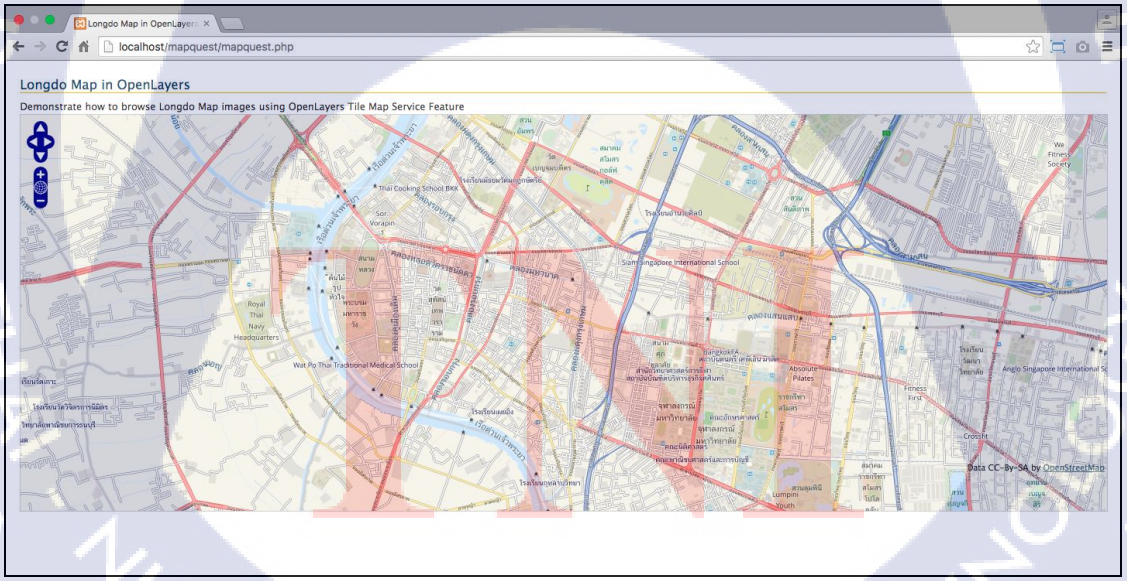
บทสรุป

ในปัจจุบันมีผู้ให้บริการให้บริการแผนที่เป็นจำนวนมาก ซึ่งส่วนใหญ่ใช้ข้อมูลจากเว็บไซต์ที่เปิดให้บริการแก้ไขพิกัดและข้อมูลแผนที่ ซึ่งผู้ให้บริการแผนที่ในหลายๆประเทศล้วนนำข้อมูลนำข้อมูลเหล่านั้นมาใช้กับแผนที่ของตนเอง เพื่อแสดงรายละเอียด เส้นทาง สถานที่ การจารจร พร้อมทั้งปรับแต่งให้มีความสวยงาม ในส่วนของการปรับแต่งรายละเอียดแผนที่เช่น ถนน เส้นทางพิเศษ ทางด่วน อาคารต่างๆนั้นสามารถปรับแต่งได้ผ่านไฟล์ที่ใช้กำหนดรูปแบบของแผนที่ ประเทศไทยเองก็มีผู้ให้บริการแผนที่ ซึ่งได้นำรูปแบบแผนที่ของประเทศอังกฤษมาปรับใช้สำหรับประเทศไทย แต่เนื่องจากยังคงต้องมีการปรับแต่งอีกมาก การทำงานผ่านเซิร์ฟเวอร์ จึงใช้เวลานาน และไม่เหมาะแก่การปรับแต่งที่มีการปรับเปลี่ยนอยู่ตลอดเวลา

โครงงานนี้ได้ศึกษาและติดตั้งเครื่องเร็นเดอร์แผนที่ เพื่อให้สามารถทำงานได้โดยไม่ต้องผ่านเซิร์ฟเวอร์ และตั้งค่าให้รองรับภาพความละเอียดสูง ผลของโครงงานนี้ได้ออกมาเป็นไฟล์ที่ใช้สำหรับเร็นเดอร์แผนที่ เว็บเพจสำหรับแสดงแผนที่ และไฟล์สคริปต์ที่ใช้อัพเดทข้อมูลแผนที่ในฐานข้อมูล โดยทั้งหมดสามารถทำงานได้บนระบบปฏิบัติการลีนุกซ์



รูปที่ 1 โค๊ดหน้าเว็บที่ใช้แสดงแผนที่ที่ถูกเร็นเดอร์ออกมา



รูปที่ 2 หน้าเว็บที่ใช้แสดงแผนที่

กิตติกรรมประกาศ

ขอขอบพระคุณ บริษัท เมตามีเดีย เทคโนโลยี จำกัด ที่ให้โอกาสฝึกฝนทำโครงงาน ทำให้ได้เรียนรู้ปฏิบัติ และได้รับประสบการณ์การในการแก้ไขปัญหาต่างๆ คุณภัทระ เกียรติเสวี บริษัท เมตามีเดีย เทคโนโลยี จำกัด ที่มอบหมายงาน พร้อมทั้งให้ข้อมูลที่เป็นความรู้และมีประโยชน์ต่อการทำโครงงาน รวมไปถึงสถานที่ที่ใช้ทำงาน คุณชุติภัทร์ โชคพิพัฒน์พร บริษัท เมตามีเดีย เทคโนโลยี จำกัด ที่คอยดูแล ให้ความรู้ ความช่วยเหลือ ตลอดเวลาที่ฝึกงาน

คุณภานุพงศ์ ชลไพรพิมลรัตน์ นักศึกษาที่ร่วมทำโครงงาน ที่คอยให้คำแนะนำเกี่ยวกับการแก้ไขปัญหาตอนติดตั้ง รวมไปถึงช่วยทดลองการทำงานบนระบบปฏิบัติการ Ubuntu อาจารย์ อมรพันธ์ ชมกลิ่น อาจารย์ที่ปรึกษาตลอดช่วงระยะเวลาในการฝึกงาน ที่ได้ให้คำปรึกษาสำหรับการจัดทำรายงาน และช่วยตรวจสอบแก้ไข จนรายงานฉบับนี้เสร็จสมบูรณ์

นาย พงศกร อยู่สุข

ผู้จัดทำ

สารบัญ

บทสรุป ..... ข
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญรูป ..... ช
สารบัญตาราง ..... ซ

บทที่

บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์โครงงาน 1
1.3 ขอบเขตของโครงงาน 1
1.4 นิยามศัพท์เฉพาะ 1
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน 3
2.1 เครื่องมือ Mapnik 3
2.1.1 Mapnik Styles 3
2.1.2 Data Source 3
2.2 โปรแกรม PostgreSQL 4
2.2.1 PostGIS 4
2.3 ชุดเครื่องมือ GDAL/OGR 5
2.4 ชุดคำสั่ง OpenLayers 6
2.4 ภาษา Python 8

สารบัญ(ต่อ)

บทที่ 3 แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน 9

3.1 แผนงานการฝึกงาน 9

3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน 9

3.2.1 แก้ไขหน้าเว็บไซต์ให้เป็น Responsive 9

3.2.2 เขียนสคริปต์แปลงรูปภาพเพื่อแก้ไขสีให้ถูกต้อง 10

3.2.3 แก้ไขสคริปต์วาดเลเซอร์ลงบนภาพ 10

3.2.4 ปรับปรุงเว็บไซต์ Mapdemo Longdo ในส่วนของ Openlayers 10

3.2.5 ติดตั้งเครื่องมือเร็นเดอร์แผนที่บนระบบปฏิบัติการ Linux และปรับแต่งให้รองรับภาพ 10

แบบเอชดี 10

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน 10

บทที่ 4 สรุปผลการดำเนินงาน 14

4.1 ผลการดำเนินงาน 14

บทที่ 5 บทสรุปและข้อเสนอแนะ 16

5.1 สรุปผลการดำเนินงาน 16

5.2 แนวทางการแก้ไขปัญหา 16

5.3 ข้อเสนอแนะ 16

เอกสารอ้างอิง 17

ประวัติผู้จัดทำโครงงาน 18

สารบัญรูป

รูป **หน้า**

รูปที่ 1 โค๊ดหน้าเว็บที่ใช้แสดงแผนที่ที่ถูกเร็นเดอร์ออกมา ..... ค
รูปที่ 2 หน้าเว็บที่ใช้แสดงแผนที่ ..... ค
รูปที่ 2.1 เว็บไซต์ทางการของ GDAL/OGR 5
รูปที่ 2.2 ตัวอย่าง Source Code จากการประมวลผลของ Plugin 7
รูปที่ 3.1 ติดตั้ง Mapnik และตรวจสอบ Python Binding 11
รูปที่ 3.2 ตั้งค่าการ Render ในไฟล์ generated_tile.py 12
รูปที่ 3.3 ข้อมูลพิกัดต่างๆของแผนที่ ถูกนำขึ้นไปไว้บนฐานข้อมูล 12
รูปที่ 3.4 หน้าเว็บที่ใช้แสดงแผนที่ ซึ่งเรียกใช้ผ่านตัว Openlayers 13
รูปที่ 3.5 สคริปต์สำหรับอัพเดทฐานข้อมูล 13
รูปที่ 4.1 Style Sheet ที่ใช้ในการ Render 14
รูปที่ 4.2 หน้าเว็บเพจที่ใช้ในการแสดงผลแผนที่ 15
รูปที่ 4.3 สคริปต์ที่ใช้ในการอัพเดทข้อมูลแผนที่ในฐานข้อมูล 15

สารบัญตาราง

ตาราง **หน้า**

ตารางที่ 3.1 แผนงานการฝึกงาน 9

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันมีผู้ให้บริการให้บริการแผนที่เป็นจำนวนมาก ซึ่งส่วนใหญ่ใช้ข้อมูลจากเว็บไซต์ที่เปิดให้บริการแก้ไขพิกัดและข้อมูลแผนที่ ซึ่งผู้ให้บริการแผนที่ในหลายๆประเทศล้วนนำข้อมูลนำข้อมูลเหล่านั้นมาใช้กับแผนที่ของตนเอง เพื่อแสดงรายละเอียด เส้นทาง สถานที่ การจารจร พร้อมทั้งปรับแต่งให้มีความสวยงาม ในส่วนของการปรับแต่งรายละเอียดแผนที่เช่น ถนน เส้นทางพิเศษ ทางด่วน อาคารต่างๆนั้นสามารถปรับแต่งได้ผ่านไฟล์ที่ใช้กำหนดรูปแบบของแผนที่ ประเทศไทยเองก็มีผู้ให้บริการแผนที่ ซึ่งได้นำรูปแบบแผนที่ของประเทศอังกฤษมาปรับใช้สำหรับประเทศไทย แต่เนื่องจากยังคงต้องมีการปรับแต่งอีกมาก การทำงานผ่านเซิร์ฟเวอร์ จึงใช้เวลานาน และไม่เหมาะแก่การปรับแต่งที่มีการปรับเปลี่ยนอยู่ตลอดเวลา

1.2 วัตถุประสงค์โครงงาน

1) เพื่อติดตั้งเครื่องมือเร็นเดอร์แผนที่ให้สามารถทำงานบนระบบปฏิบัติการลีนุกซ์ได้
2) เพื่อตั้งค่าการเร็นเดอร์ให้รองรับภาพที่มีความละเอียดสูง
3) เพื่อปรับแต่งแผนที่ได้ตามความต้องการโดยไม่ต้องทำงานผ่านเซิร์ฟเวอร์

1.3 ขอบเขตของโครงงาน

1) ศึกษาและติดตั้งเครื่องมือเร็นเดอร์แผนที่ แมพนิค
2) ตั้งค่าให้สามารถทำงานได้โดยไม่ผ่านเซิร์ฟเวอร์
3) ตั้งค่าการเร็นเดอร์ให้ที่จะนำมาใช้งานบนระบบปฏิบัติการลีนุกซ์

1.4 นิยามศัพท์เฉพาะ

การเร็นเดอร์ (Render) หรือ การสร้างภาพกราฟิกส์ บางครั้งเรียกว่า "การสร้างภาพจากแบบจำลอง" หรือ "การสร้างเป็นภาพสุดท้าย" คือกระบวนการสร้างภาพสองมิติจากแบบจำลองกราฟิกในระบบ โดยเริ่มจากการนำเข้าแบบจำลองกราฟิกซึ่งจะบรรยายวัตถุสองมิติ หรือสามมิติโดยบอกโครงสร้าง

ข้อมูลของวัตถุสามมิติ อันประกอบด้วยข้อมูลเชิงเรขาคณิต ได้แก่พิกัด มุมมอง พื้นผิวลวดลาย และข้อมูลเกี่ยวกับความสว่าง และคำนวณเพื่อแสดงผลลัพธ์เป็นภาพสองมิติบนจอ ซึ่งจะเป็นภาพแบบดิจิทัล (ภาพเชิงเลข) หรือภาพแบบจุดภาพ (ภาพแรสเตอร์) ทั้งนี้การเร็นเดอร์ภาพไม่ได้จำกัดอยู่เฉพาะการให้แสงและเงา ในบางกรณีก็หมายถึงการให้สีหรือการให้เส้น โดยไม่ต้องมีการให้แสงเงาก็ได้

ลีนุกซ์ (Linux) และรู้จักในชื่อ กะนู/ลินุกซ์ (GNU/Linux) โดยทั่วไปเป็นคำที่ใช้ในความหมายที่หมายถึงระบบปฏิบัติการแบบยูนิกซ์ โดยใช้ลีนุกซ์ เคอร์เนล เป็นศูนย์กลางทำงานร่วมกับไลบรารีและเครื่องมืออื่น ลินุกซ์เป็นตัวอย่างหนึ่งในฐานะซอฟต์แวร์เสรี และซอฟต์แวร์ โอเพนซอร์สที่ประสบความสำเร็จและมีชื่อเสียง ทุกคนสามารถดูหรือนำโค้ดของลีนุกซ์ไปใช้งาน, แก้ไข, และแจกจ่ายได้อย่างเสรี ลีนุกซ์นิยมจำหน่ายหรือแจกฟรีในลักษณะเป็นแพคเกจ โดยผู้จัดทำจะรวมซอฟต์แวร์สำหรับใช้งานในด้านอื่นเป็นชุดเข้าด้วยกัน

แมพนิค (Mapnik) เป็นเครื่องมือที่ใช้สำหรับเร็นเดอร์แผนที่ ใช้งานได้ผ่านภาษา C++ และ Python เพื่อรองรับการพัฒนาแบบรวดเร็ว สามารถใช้งานได้ทั้งการออกแบบแผนที่บนคอมพิวเตอร์และการพัฒนาบนเว็บไซต์ สามารถอ่านไฟล์ได้หลายชนิด และสามารถเร็นเดอร์แผนที่ออกเป็นไฟล์รูปนามสกุลต่างๆได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถติดตั้งเครื่องมือเร็นเดอร์แผนที่ ที่สามารถทำงานได้บนบนระบบปฏิบัติการลีนุกซ์
2) มีเครื่องมือรองรับการทำงานกับภาพที่มีความละเอียดสูง
3) สามารถปรับแต่งแผนที่สะดวกและรวดเร็วมากขึ้น เพราะไม่ต้องทำงานผ่านเซิร์ฟเวอร์

บทที่ 2
ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

เทคโนโลยีที่ใช้ส่วนใหญ่ทำงานได้ในทุกระบบปฏิบัติการ ในโครงงานนี้ใช้ระบบปฏิบัติการ OSX ในการทำงาน ซึ่งมีการทำงานเหมือนกับบน Linux และได้นำ Mapnik, PostgresSQL, GDAL, OpenLayers และ Python มาใช้ในการเร็นเดอร์แผนที่

2.1 เครื่องมือ Mapnik

Mapnik เป็นเครื่องมือที่ใช้สำหรับเร็นเดอร์แผนที่ ใช้งานได้ผ่านภาษา C++ และ Python เพื่อรองรับการพัฒนาแบบรวดเร็ว สามารถใช้งานได้ทั้งการออกแบบแผนที่บนคอมพิวเตอร์และการพัฒนาบนเว็บไซต์ Mapnik สามารถอ่าน Shapefiles, PostGIS, TIFF rasters, ไฟล์นามสกุล .osm และไฟล์ที่รองรับรูปแบบของ GDAL หรือ OGR และสามารถเร็นเดอร์แผนที่ออกเป็นไฟล์รูปนามสกุลต่างๆ เช่น PNG, JPEG, SVG หรือ PDF เป็นต้น รวมไปถึงยังมีเว็บไซต์ที่ให้ศึกษาตัวเครื่องมืออย่างครบถ้วน

2.1.1 Mapnik Styles

Mapnik อนุญาตให้ปรับแต่งขนาดแผนที่ รวมไปถึงสัญลักษณ์ สี รูปแบบตัวหนังสือ ข้อมูลต่างๆ อย่างอิสระ ผ่านแหล่งข้อมูลและ Style Sheet ส่วนใหญ่เป็นภาษา XML นอกเหนือจากนี้ยังมีเครื่องมือเสริมตัวอื่นที่ช่วยในการปรับแต่ง Style เช่น Cascadenik, Spreadnik และ TileMill ที่มีการใช้งานที่ง่ายกว่า XML ของ Mapnik

2.1.2 Data Source

แหล่งข้อมูลของ Mapnik นั้นมาจากหลายที่ ผ่านทางข้อมูล OSM โดยตรง จากฐานข้อมูล PostGIS หรือ shapfiles เป็นต้น

2.1.2.1 PostGIS

PostGIS เป็นวิธีดึงข้อมูล OSM ที่นิยมที่สุด โดยสามารถนำข้อมูลขึ้นไปบนฐานข้อมูลได้ผ่าน เครื่องมือ osm2pgsql หรือ Imposm และเข้าถึงผ่าน SQL Queries และ GIS Funtions ที่ถูกกำหนดผ่านตัว Mapnik Style โดยการดึงข้อมูลผ่านทางนี้สามารถนำไปประยุกต์ใช้ได้หลากหลาย รวมไปถึงยังเป็นแหล่งข้อมูลหลักในการเร็นเดอร์ตัว OSM layers

2.1.2.2 Shapefile

Shapefile เป็นไฟล์ที่ใช้เก็บข้อมูลทางภูมิศาสตร์ ซึ่งนอกเหนือจาก PostGIS แล้ว Style บางตัวใช้ shapefile ในการเร็นเดอร์แผนที่เช่นกัน

2.2 โปรแกรม PostgreSQL

PostgreSQL เป็นระบบการจัดการฐานข้อมูลเชิงวัตถุ-สัมพันธ์ (object-relational) แบบ ORDBMS โดยสามารถใช้รูปแบบคำสั่งของภาษา SQL ได้เกือบทั้งหมด นอกจากนี้ยังเป็นระบบฐานข้อมูลที่ทันสมัยที่สุดของ OpenSource ที่สามารถนำไปใช้งานได้โดยไม่มีค่าใช้จ่ายใด ได้มีการพัฒนามาจาก POSTGRES 4.2 โดยมหาวิทยาลัยแคลิฟอร์เนีย (Berkeley Computer Science department, University of California.)

PostgreSQL ชื่อเดิมคือ Postgres เป็น DBMS เป็นระบบจัดการฐานข้อมูลแบบ object-relational database management system หรือ (ORDBMS) พัฒนาที่ University of California, Berkeley (มหาวิทยาลัย) เริ่มโครงการโดย Michael Stonebraker เมื่อปี พ.ศ. 2528 ในยุคแรกชื่อของระบบเรียกว่า post-Ingres เนื่องมาจากจากเป็นระบบที่มีวิวัฒนาการมาจากระบบจัดการฐานข้อมูล Ingres

คุณสมบัติสำคัญของ PostgreSQL คือการมีคุณสมบัติ ACID ((Atomicity, Consistency, Isolation, Durability) ครบถ้วนโดยสนับสนุน foreign keys, joins, views, triggers, และ stored procedures (หลายภาษา) โดยมีชนิดข้อมูลใน SQL92 และ SQL99 ได้แก่ INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, และ TIMESTAMP

นอกจากนี้ PostgreSQL ยังทำงานในหลายแพลทฟอร์มได้แก่ Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), และ Windows

ลักษณะสำคัญอีกอย่างหนึ่งคือ PostgreSQL เป็นซอฟต์แวร์แบบรหัสเปิดใช้ลิขสิทธิ์ BSD ซึ่งหมายถึงผู้ใช้สามารถนำไปใช้งานได้ฟรี นอกจากนี้ในปัจจุบัน PostgreSQL ไม่อยู่ภายใต้การควบคุมขององค์กรใดโดยเฉพาะแต่มีผู้ร่วมพัฒนาจากทั่วโลกทำให้ PostgreSQL มีการปรับปรุงอย่างต่อเนื่อง

2.2.1 PostGIS

PostGIS เป็นฐานข้อมูล ด้านแผนที่ภูมิศาสตร์ โดยจะเป็นฐานข้อมูลที่มากับ ฐานข้อมูล PostgreSQL ซึ่งเป็น open source โดยการทำงานจะทำงานระดับ Geomatry ซึ่งความสามารถจะด้อยกว่าที่ ArcSDE for PostgreSQL ซึ่งการทำงานจะเท่าเทียบกับ Oracle Spacilist

2.3 ชุดเครื่องมือ GDAL/OGR

GDAL/OGR เป็นชุดเครื่องมือจัดการและประมวลผลข้อมูลราสเตอร์และเวกเตอร์ พัฒนาขึ้นด้วยภาษา C / C++ มีความสามารถรองรับและสนับสนุนรูปแบบของข้อมูลภูมิสารสนเทศทั้งเวกเตอร์และราสเตอร์หลายรูปแบบทั้งการอ่าน/แก้ไข/และการเขียน ปัจจุบันมีผู้นำ GDAL/OGR ไปใช้ในซอฟต์แวร์ประยุกต์สำหรับประมวลผลภูมิสารสนเทศทั้งในซอฟต์แวร์เชิง พาณิชย์ เช่น ArcGIS, MapGuide, Google Earth, Cadcorp SIS, FME และซอฟต์แวร์รหัสเปิด เช่น GRASS, QGIS, GeoServer และ OpenEV เป็นต้น และยังได้มีการพัฒนาขยายขีดความสามารถให้นำไปใช้ได้กว้างขวางขึ้นโดยใช้ร่วม กับภาษาอื่นในลักษณะ Binding เช่น ใช้ได้กับภาษา Python, Java, C#, Ruby, VB6 และ Perl สถานะของชุดเครื่องมือเป็นโปรแกรมรหัสเปิด อนุญาตให้ใช้ตามสัญญา X/MIT

GDAL/OGR แบ่งออกเป็น2 ส่วน ส่วนแรกคือ GDAL (Geo-spatial Data Abstraction Library) มีขีดความสามารถครอบคลุมด้านการประมวลผลข้อมูลราสเตอร์ ส่วนที่สองคือOGR library ครอบคลุมการใช้งานกับข้อมูลเวกเตอร์ ทั้งสองส่วนเมื่อผนวกกันไว้ ทำให้เป็นชุดเครื่องมือระดับแกนกลางที่นักพัฒนาโปรแกรมประยุกต์สามารถนาไป ใช้ประโยชน์ได้เป็นอย่างดีการเรียกขานชื่อ GDAL/OGR เนื่องจากในระยะต้นเป็นเพียงชุดเครื่องมือด้านข้อมูลราสเตอร์เพียงอย่าง เดียว เป็นที่รู้จักในชื่อGDAL หลังจากได้รวมเอาส่วนของOGR ไว้ด้วย จึงเรียกว่า GDAL/OGR แต่ถ้าเจาะจงใช้ชื่อGDAL ให้หมายถึงกรณีที่กล่าวถึงบริบทที่เป็นเรื่องการทำงานกับข้อมูลแบบราสเตอร์ และในทำนองเดียวกัน เมื่อกล่าวถึงชื่อ OGR ก็ให้หมายถึงบริบทของการทำงานกับข้อมูลแบบเวกเตอร์

รูปที่ 2.1 เว็บไซต์ทางการของ GDAL/OGR

ชุด เครื่องมือ GDAL/OGR จะประกอบด้วย Library ของเครื่องมือต่างๆ ในรูปแบบของภาษา C/C++ ส่วนดังกล่าวนี้เองที่ผู้พัฒนาโปรแกรมสามารถนาไปใช้เป็นเครื่องมือประมวล หลักต่างๆ ในซอฟต์แวร์หรือโปรแกรมประยุกต์ของตนเอง ดังเห็นได้จาก GDAL/OGR ถูกนำไปใช้เป็นเครื่องมือประมวลผลของซอฟต์แวร์ต่างๆ ไม่ว่าจะเป็นซอฟต์แวร์เชิงพาณิชย์และซอฟต์แวร์รหัสเปิดดังที่กล่าวข้างต้น นอกเหนือจากการนำไปใช้โดยตรง มีกลุ่มผู้พัฒนาโปรแกรมหลายกลุ่มได้พัฒนา GDAL/OGR ให้สามารถใช้ได้กับโปรแกรมภาษาอื่นๆ โดยเรียกใช้จากสภาพแวดล้อมของการทำงานของโปรแกรมภาษานั้นๆ ด้วยการเขียนโปรแกรมครอบตัวเครื่องมือคำสั่งหลักของ GDAL/OGR อีกชั้นหนึ่ง เช่น Python, Java, C#, Ruby, VB6 และ Perl สิ่งนี้เองทำให้สามารถขยายการใช้งานจากกลุ่มนักพัฒนาโปรแกรมภาษา C/C++ ไปยังกลุ่มผู้พัฒนาด้วยโปรแกรมภาษาอื่นๆ ได้เป็นอย่างดีการใช้งาน GDAL/OGR ทำได้อีกลักษณะสำหรับผู้ใช้งานทั่วไป คือการพิมพ์คำสั่งเรียกใช้เครื่องมือต่างๆ ที่มีของ GDAL/OGR ในลักษณะของการพิมพ์คำสั่ง (Command Line) โดยที่ GDAL/OGR ได้ทำการคอมไพล์ชุดเครื่องมือหลักขึ้นจานวนหนึ่งเป็นไฟล์ที่สามารถสั่งทำงาน ได้ทันที (Executable File) โดยแบ่งเป็นชุดเครื่องมือ

GDAL และ OGR ทั้ง 2 ส่วนจะประกอบด้วยเครื่องมือการแปลงรูปแบบไฟล์ข้อมูลจากรูปแบบหนึ่งไปอีกรูป แบบหนึ่ง ซึ่งชนิดของรูปแบบข้อมูลมีการสนับสนุนอยู่เป็นจำนวนมาก ทั้งราสเตอร์และเวกเตอร์ นอกจากนี้ยังสนับสนุนการแปลงค่าระบบพิกัดต่างๆ ตามมาตรฐานสากลที่มีอยู่ หรือจากการกำหนดค่าพารามิเตอร์เฉพาะตามแต่พื้นที่ได้เอง ด้วยความสามารถและการใช้งานในลักษณะนี้ จึงเพิ่มความสะดวกให้กับผู้ใช้งานทั่วไปที่ต้องการแปลงข้อมูลไปสู่รูปแบบ ต่างๆ รวมถึงการปรับค่าระบบพิกัดได้ด้วยตนเอง ด้วยการพิมพ์คำสั่งที่มีอยู่ ตามความต้องการของผู้ใช้งาน อย่างไรก็ตามการใช้งานดังกล่าวเป็นลักษณะของการพิมพ์คำสั่ง ผู้ใช้ต้องจำคำสั่งได้หรือสามารถเรียกดูข้อแนะนำการใช้งานของแต่ละโปรแกรมได้เอง ซึ่งไม่มีลักษณะการทำงานแบบเมนูให้ไว้ ดังนั้นผู้ใช้งานที่ต้องการใช้ผ่านทางเมนูจึงต้องหาซอฟต์แวร์ที่มีเครื่อง มือดังกล่าวเรียกใช้งานเอง เช่น QGIS เป็นต้น

2.4 ชุดคำสั่ง OpenLayers

OpenLayers เป็นชุดคำสั่ง Java Script ที่พัฒนาขึ้นเพื่อเป็นทางเลือกสำหรับการพัฒนาโปรแกรมด้านภูมิสารสนเทศบนเว็บไซต์ ฝั่ง Web Client นอกเหนือจาก Google Maps API สนับสนุนการเชื่อมต่อกับระบบให้บริการข้อมูลภูมิสารสนเทศที่หลากหลาย เช่น WMS, WFS, WMTS, Google, WorldWind, Yahoo, MultiMap, TileCache, MapGuide, ArcIMS และ ArcGIS93Rest เป็นต้น ซึ่งมีเครื่องมือควบคุมการแสดงผล

ต่างๆ จำนวนมาก ไม่ว่าจะเป็น การ zoom/pan, การหาตำแหน่งจากตัวชี้ตำแหน่ง, มาตราส่วน, เครื่องมือควบคุมการเปิด/ปิดการแสดงผล, เครื่องมือการวาดรูป เหล่านี้ เป็นต้น

OpenLayers ถูกนำไปประยุกต์ใช้และต่อยอดจำนวนมาก ในหลายๆ ซอฟต์แวร์และชุดคำสั่งประยุกต์รหัสเปิด (Open Source) เช่น เป็นเครื่องมือสำหรับแสดงผลข้อมูลใน GeoServer, เป็นเครื่องมือสำหรับต่อเชื่อมและแสดงข้อมูล Google ใน QGIS เป็นต้น และยังถูกนำไปใช้ใน WebSite ต่างๆ ที่เผยแพร่ข้อมูลภูมิสารสนเทศ ทั่วโลกรวมถึงในประเทศไทย ทั้งในลักษณะที่เป็นโปรแกรมประยุกต์เฉพาะเรื่อง หรือเป็นองค์ประกอบหนึ่งใน Web Page อีกด้วย ซึ่งมีข้อดี ดังนี้

- ทำให้การพัฒนา Web ในการเผยแพร่ข้อมูล GIS ง่ายมากขึ้น
- เป็น Open source Software ที่ผู้ใช้สามารถ Download มาใช้ได้ฟรี โดยไม่มีค่าใช้จ่าย ทำให้เครื่องมือดังกล่าวนี้เป็นที่นิยมสำหรับนักพัฒนา Web GIS
- พัฒนาโดยใช้ภาษา JavaScript / AJAX (Asynchronous JavaScript and XML)
- เหมาะสำหรับการเริ่มต้นพัฒนาระบบ Web GIS

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>OGR2Layers</title>
<style>
 #map{width:400px;height:400px;}
</style>
<script src="http://www.openlayers.org/api/2.6/OpenLayers.js"></script>
<script type="text/javascript">
    var map
    function init(){
        map = new OpenLayers.Map('map');
        var olwms =  new OpenLayers.Layer.WMS( "OpenLayers WMS", ["http://labs.metacarta.com/wms/vmap0"],
                {layers: "basic", format: "image/gif" } );
        map.addLayer(olwms);
        map.addLayer(new OpenLayers.Layer.GML("amphoe KML","amphoe.kml",
                {format: OpenLayers.Format.KML}));
        var ls= new OpenLayers.Control.LayerSwitcher();
        map.addControl(ls);
        ls.maximizeControl();
        map.zoomToExtent(new OpenLayers.Bounds(99.4649895548,13.5148896009,100.879454684,14.3183727514))
    };
</script>
</head>
<body onload="init()">
<h1>Province</h1>
<div id="map"></div>
</body>
</html>
```

รูปที่ 2.2 ตัวอย่าง Source Code จากการประมวลผลของ Plugin

2.4 ภาษา Python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัว นี้เป็น OpenSource เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีๆโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครบคุมกับทุกลักษณะงาน

โค้ดของ Python ถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์ คือจะประมวลผลไปทีละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ Python โดยมีจุดเด่นของภาษา Python ดังนี้

- Python เป็นภาษาสคริปต์ ทำให้ใช้เวลาในการเขียนและคอมไพล์ไม่มาก ทำให้เหมาะกับงานด้านการดูแลระบบ (System administration) เป็นอย่างยิ่ง ได้มีการสนับสนุนภาษา Pythonโดยเป็นส่วนหนึ่งของระบบปฏิบัติการยูนิกซ์, ลินุกซ์ และสามารถติดตั้งให้ทำงานเป็นภาษาสคริปต์ของวินโดวส์ผ่านระบบ Windows Script Host ได้อีกด้วย
- ไวยากรณ์ของ Pythonได้กำจัดการใช้สัญลักษณ์ที่ใช้ในการแบ่งบล็อกของโปรแกรม และใช้การย่อหน้าแทน ทำให้สามารถอ่านโปรแกรมที่เขียนได้ง่าย นอกจากนั้นยังมีการสนับสนุนการเขียน docstring ซึ่งเป็นข้อความสั้นๆ ที่ใช้อธิบายการทำงานของฟังก์ชัน, คลาส, และโมดูลอีกด้วย
- Python เป็นภาษากาว (Glue Language) ได้อย่างดีเนื่องจากสามารถเรียกใช้ภาษาโปรแกรมอื่นๆ ได้หลายภาษา ทำให้เหมาะที่จะใช้เขียนเพื่อประสานงานโปรแกรมที่เขียนในภาษาต่างกันได้

บทที่ 3
แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

ในส่วนนี้เป็นแผนการปฏิบัติงานและขั้นตอนในการดำเนินงานของโครงงานทั้งหมด โดยแบ่งเป็นหัวข้อ ดังนี้

3.1 แผนงานการฝึกงาน

ตารางที่ 3.1 แผนงานการฝึกงาน

หัวข้อ	เดือนที่ 1				เดือนที่ 2			
1. แก้ไขหน้าเว็บไซต์ให้เป็น Responsive								
2. เขียนสคริปต์แปลงรูปภาพเพื่อแก้ไขสีให้ถูกต้อง								
3. แก้ไขสคริปต์วาดเลเซอร์ลงบนภาพ								
4. ปรับปรุงเว็บไซต์ Mapdemo Longdo ในส่วนของ Openlayers								
5. ติดตั้งเครื่องมือเร็นเดอร์แผนที่บนระบบปฏิบัติการ Linux และปรับแต่งให้รองรับภาพแบบเอชดี								

3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน

3.2.1 แก้ไขหน้าเว็บไซต์ให้เป็น Responsive

ทำการปรับปรุงหน้าเว็บ Longdo Traffic ให้เป็น Responsive เว็บไซต์ ตั้งแต่หน้าจอโทรศัพท์มือถือไปจนถึงคอมพิวเตอร์ตั้งโต๊ะ ระยะเวลา 1 สัปดาห์

3.2.2 เขียนสคริปต์แปลงรูปภาพเพื่อแก้ไขสีให้ถูกต้อง

จัดทำสคริปต์ที่ทำงานบนระบบปฏิบัติการ Linux เพื่อใช้ในการแปลงรูปภาพแผนที่ เพื่อแก้ไขสีที่ผิดเพี้ยนให้เป็นปกติ เพื่อนำไปใช้งานบนเว็บไซต์จริง

3.2.3 แก้ไขสคริปต์วาดเลเซอร์ลงบนภาพ

ทำการแก้ไขสคริปต์ที่ใช้สำหรับวาดเลเซอร์ลงบนภาพ ให้สามารถรับค่ารัศมีในการวาดเลเซอร์ และกำหนดสีให้กับเลเซอร์ที่วาดได้ โดยนำสคริปต์ที่มีอยู่แล้วมาทำการแก้ไข และทำให้สามารถเรียกใช้ได้จากคำสั่งเดียว

3.2.4 ปรับปรุงเว็บไซต์ Mapdemo Longdo ในส่วนของ Openlayers

ทำปรับปรุงเว็บไซต์ Mapdemo ของ Longdo ในส่วนของ Openlayers ให้แสดงผลให้ถูกต้องจากนั้นทำการอัพเกรทจาก Openlayers เวอร์ชัน 2 ให้เป็น เวอร์ชัน 3 เพื่อนำไปใช้งานบนเว็บไซต์สำหรับผู้ใช้ที่ต้องการดูตัวอย่างการนำแผนที่ของ Longdo ไปใช้งาน

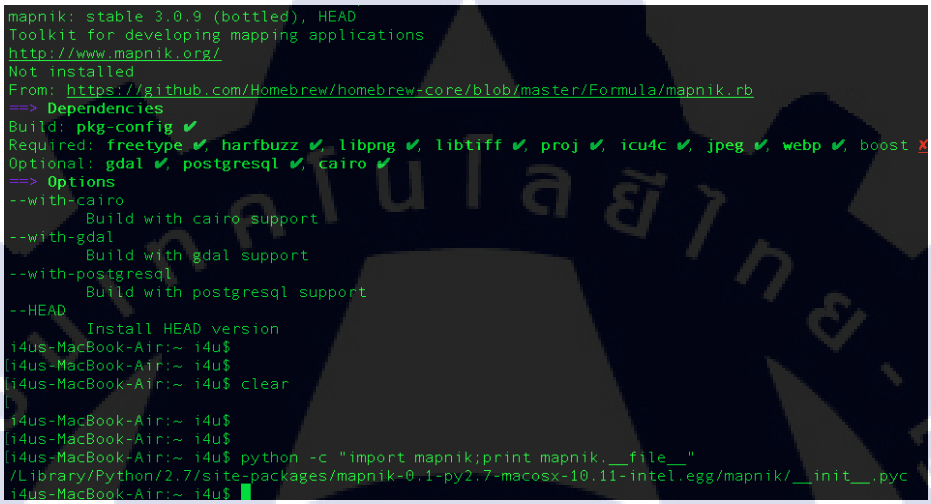
3.2.5 ติดตั้งเครื่องมือเร็นเดอร์แผนที่บนระบบปฏิบัติการ Linux และปรับแต่งให้รองรับภาพแบบเอชดี

ทำการติดตั้งเครื่องมือเร็นเดอร์แผนที่ที่สามารถนำไปใช้งานบนระบบปฏิบัติการ Linux พร้อมทั้งสามารถปรับแต่งรูปแบบของแผนที่ ความละเอียดของภาพที่ทำเร็นเดอร์ เพื่อใช้ในการออกแบบ style ของแผนที่เพื่อปรับปรุงจากของเดิม โดยที่สามารถแก้ไขและแสดงผลผ่านเครื่องคอมพิวเตอร์ส่วนตัวได้ ไม่ต้องทำงานผ่าน Server เพื่อความสะดวกรวดเร็ว

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน

1. ศึกษาเครื่องมือที่ใช้ในการ Render Mapnik ในขั้นตอนแรกจะทำการศึกษาเครื่องมือที่ใช้สำหรับ Render แผนที่ ในที่นี้ใช้ Mapnik ในการ Render ซึ่งได้ทำการศึกษาวิธีการติดตั้ง ความต้องการของระบบ วิธีการใช้งาน และปัญหาที่อาจเกิดขึ้น

2. เตรียมทรัพยากรเครื่องและติดตั้ง Mapnik ขั้นตอนนี้เป็นการเตรียมทรัพยากรเครื่องและติดตั้ง Mapnik โดยจะต้องทำการลง Library ต่างๆ ดังนี้ Python สำหรับใช้ในการ Render , Libxml2 สำหรับอ่าน XML Style Sheet, proj สำหรับการตั้งค่า Projection ของแผนที่, GDAL สำหรับการจัดการรูปภาพและ Projection, PostgresSQL ใช้เป็นฐานข้อมูล และ osm2pgsql ไว้ใช้สำหรับดึงข้อมูล OSM ขึ้นฐานข้อมูล จากนั้นทำการติดตั้ง Mapnik พร้อมทดสอบการ Binding Python เพื่อตรวจสอบการใช้งาน Python



รูปที่ 3.1 ติดตั้ง Mapnik และตรวจสอบ Python Binding

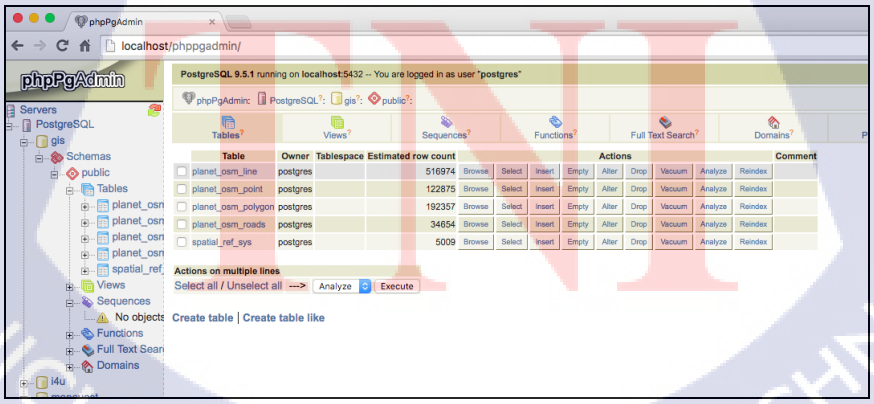
3. ตั้งค่าการ Render ให้กับ Mapnik ทำการโหลด Mapnik Style Sheet ซึ่งในโฟลเดอร์จะมีไฟล์ที่ใช้สำหรับ Render แผนที่ ชื่อว่า generated_tile.py โดยการทำงานมันจะไปดึงค่าจากไฟล์ XML Style Sheet มา Render ออกมาเป็น Tile ของแผนที่ ซึ่งในไฟล์นี้ค่าโดยเริ่มต้นจะเป็นขนาด 256x256 จึงได้ทำการแก้ไขให้เป็น 512x512 และทำการระบุที่อยู่ของไฟล์ Style Sheet ที่จะใช้ในการ Render พร้อมทั้งระบุที่อยู่ของไฟล์รูปภาพที่จะทำการ Render ออกมา ซึ่งในไฟล์ generated_tile.py นี้สามารถระบุขอบเขตของพื้นที่ ที่จะทำการ Render ได้ ในที่นี้จะใช้แค่ประเทศไทยเท่านั้น เนื่องจากการ Render ในช่วงระยะที่ Zoom มากๆ นั้น ใช้เวลานานจึงต้องทำการลดขอบเขตพื้นที่ลงมา

```
        self.m = mapnik.Map(512, 512)
        self.printLock = printLock
        # Load style XML
        mapnik.load_map(self.m, mapfile, True)
        # Obtain <Map> projection
        self.prj = mapnik.Projection(self.m.srs)
        # Projects between tile pixel co-ordinates and LatLong (EPSG:4326)
        self.tileproj = GoogleProjection(maxZoom+1)


    def render_tile(self, tile_uri, x, y, z):

        # Calculate pixel positions of bottom-left & top-right
        p0 = (x * 512, (y + 1) * 512)
        p1 = ((x + 1) * 512, y * 512)

        # Convert to LatLong (EPSG:4326)
        l0 = self.tileproj.fromPixelToLL(p0, z);
        l1 = self.tileproj.fromPixelToLL(p1, z);

        # Convert to map projection (e.g. mercator co-ords EPSG:900913)
        c0 = self.prj.forward(mapnik.Coord(l0[0],l0[1]))
        c1 = self.prj.forward(mapnik.Coord(l1[0],l1[1]))

        # Bounding box for the tile
        if hasattr(mapnik,'mapnik_version') and mapnik.mapnik_version() >= 800:
            bbox = mapnik.Box2d(c0.x,c0.y, c1.x,c1.y)
        else:
            bbox = mapnik.Envelope(c0.x,c0.y, c1.x,c1.y)
        render_size = 512
        self.m.resize(render_size, render_size)
        self.m.zoom_to_box(bbox)
        if(self.m.buffer_size < 256):
            self.m.buffer_size = 256

        # Render image with default Agg renderer
```

รูปที่ 3.2 ตั้งค่าการ Render ในไฟล์ generated_tile.py

4. แก้ไข Style Sheet ให้รองรับกับความละเอียด ขั้นตอนต่อมาทำการโหลด Style Sheet ของ MapQuest Open เพื่อนำมาใช้ในการ Render แผนที่ โดยหลังจากที่โหลดมาแล้ว และได้ทำการตรวจสอบพบว่า ตัวไฟล์ไม่ได้รับการปรับปรุงเป็นเวลานาน จึงต้องทำการแก้ไขโค๊ดข้างในให้ถูกต้องตาม Syntax ของ Mapnik และทำการปรับขนาดตัวอักษรให้มีขนาดมากขึ้น เนื่องจากมีการปรับขนาดความละเอียดของภาพจาก 256x256 ให้เป็น 512x512

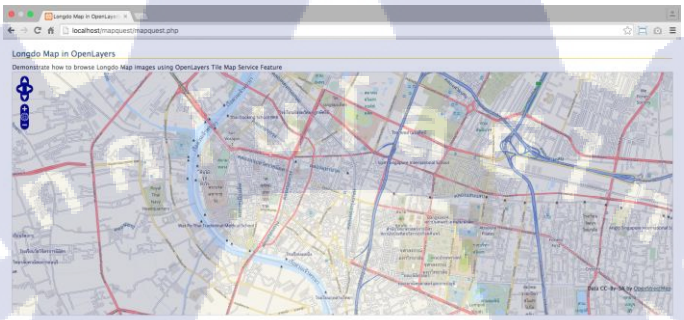
5. นำข้อมูลแผนที่ขึ้นบนฐานข้อมูลในขั้นตอนนี้จะทำการโหลดข้อมูลแผนที่ของ OSM มาเพื่อนำขึ้นไปไว้บนฐานข้อมูล โดยจะเลือกขอบเขตแค่ในประเทศไทย และใช้ osm2pgsql ในการนำข้อมูลขึ้นไปไว้บนฐานข้อมูลที่ได้มีการติดตั้งไว้ก่อนหน้านี้ พร้อมทั้งติดตั้งตัว Extension ของ PostGIS ให้กับฐานข้อมูลเพื่อใช้สำหรับแผนที่โดยเฉพาะ จากนั้นทำการตรวจสอบในฐานข้อมูลว่ามีข้อมูลถูกนำขึ้นมาแล้วหรือไม่



รูปที่ 3.3 ข้อมูลพิกัดต่างๆของแผนที่ ถูกนำขึ้นไปไว้บนฐานข้อมูล

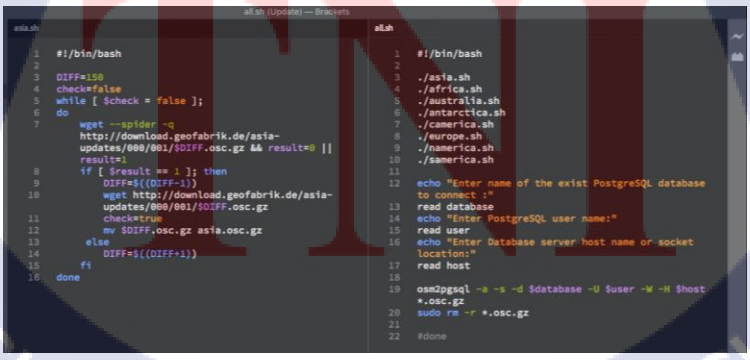
6. ทดสอบและทำการ Render เมื่อทำการตั้งค่าเสร็จแล้ว จึงทำการรันไฟล์ generated_tile.py เพื่อทดสอบการ Render แผนที่ เมื่อตรวจสอบไฟล์ที่ได้ออกมาว่าถูกต้อง จากนั้นก็ทำการ Render ภาพทั้งหมดออกมา

7. สร้างเว็บเพจเพื่อดึงแผนที่ไปแสดงผลผ่าน OpenLayers ในขั้นตอนนี้จะทำการสร้างเว็บเพจและนำไฟล์ที่ Render ได้ ออกมาแสดง โดยใช้ตัว OpenLayers ในการจัดการและแสดงแผนที่บนเว็บ เพื่อตรวจสอบผลลัพธ์ที่ได้จากการ Render โดยทั้งหมดทำบน Localhost ผ่าน Apache Web Server



รูปที่ 3.4 หน้าเว็บที่ใช้แสดงแผนที่ ซึ่งเรียกใช้ผ่านตัว Openlayers

8. ทำสคริปต์สำหรับอัพเดทฐานข้อมูลแผนที่ เมื่อติดตั้งเครื่องมือ Render และได้ทำการ Render ออกมาพร้อมทั้งแสดงผลลัพธ์บนเว็บเพจแล้ว ขั้นตอนสุดท้ายคือการอัพเดทฐานข้อมูล เนื่องจากข้อมูลแผนที่มีการอัพเดททุกวัน ทำให้ต้องโหลดไฟล์ Diff ซึ่งเก็บค่าการเปลี่ยนแปลงของข้อมูล ทางผู้จัดทำจึงได้เขียนสคริปต์ที่ทำการตรวจสอบและดาวน์โหลดข้อมูลจากนั้นทำการอัพเดทขึ้นไปบนฐานข้อมูล เป็นอันเสร็จสิ้น



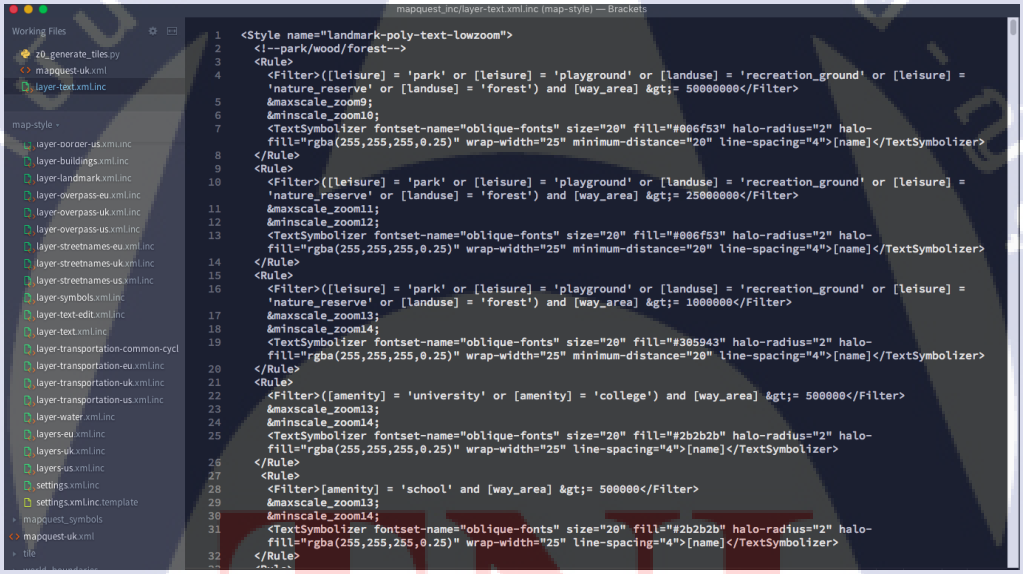
รูปที่ 3.5 สคริปต์สำหรับอัพเดทฐานข้อมูล

บทที่ 4
สรุปผลการดำเนินงาน

4.1 ผลการดำเนินงาน

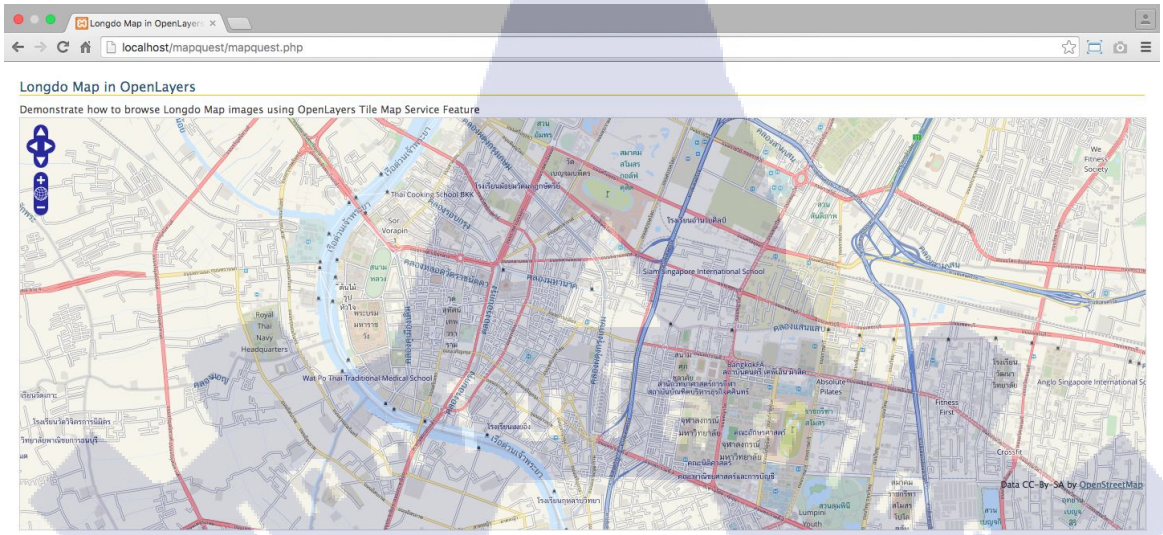
จากหัวข้อ 3.3 (ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน) วัตถุประสงค์ของโครงงานชิ้นนี้ คือการติดตั้งเครื่องมือ Render แผนที่บนระบบปฏิบัติการ Linux และปรับแต่งให้รองรับภาพแบบ HD โดยแบ่งวิธีการดำเนินงานออกเป็น 8 ขั้นตอน ซึ่งผลการดำเนินงานนั้น มีดังนี้

1) ได้ออกมาเป็นไฟล์สำหรับ Render และไฟล์ Style Sheet ที่ใช้ในการ Render



รูปที่ 4.1 Style Sheet ที่ใช้ในการ Render

2) หน้าเว็บเพจที่แสดงผลลัพธ์ของแผนที่



รูปที่ 4.2 หน้าเว็บเพจที่ใช้ในการแสดงผลแผนที่

3) สคริปต์สำหรับอัพเดทข้อมูลแผนที่ในฐานข้อมูล

รูปที่ 4.3 สคริปต์ที่ใช้ในการอัพเดทข้อมูลแผนที่ในฐานข้อมูล

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงงานชิ้นนี้ได้ทำการศึกษาเครื่องมือ Render แผนที่ พร้อมทั้ง Style Sheet สำหรับแผนที่ รวมทั้งได้มีการติดตั้ง Mapnik ควบคู่ไปกับ Style Sheet ของ MapQuest เพื่อให้ทำงานบนระบบปฏิบัติการ Linux พร้อมทั้งปรับแก้ไขความละเอียดภาพให้รองรับภาพแบบ HD (512x512) รวมไปถึงขนาดของ Font ที่แสดงบนแผนที่ และแสดงผลผ่านหน้าเว็บเพจ โดยทั้งหมดจะทำงานบนเครื่องคอมพิวเตอร์ ไม่มีการใช้งาน Server จริง โดยจะทำงานผ่าน Apache Web Server แต่เนื่องจาก Style Sheet ที่ไม่ได้รับการปรับปรุงแก้ไขเป็นเวลานาน ทำให้โค้ดส่วนมากนั้นล้าสมัย และต้องทำการแก้ไขให้ตรงกับ Style ที่ใช้อยู่ในปัจจุบัน

5.2 แนวทางการแก้ไขปัญหา

1) ทำการค้นหา Style Sheet ที่มีการปรับปรุงแก้ไขล่าสุด และอาจจะต้องติดตั้งโปรแกรมเพิ่ม เพื่อใช้งาน Style Sheet เหล่านั้น
2) ปรับปรุงแก้ไข Style Sheet ที่มี ให้เป็นของเราเอง อาจจะใช้เวลามากขึ้น
3) การ Render แต่ละรอบนั้นใช้เวลานาน ถ้าเครื่องคอมพิวเตอร์ที่ประสิทธิภาพไม่สูงมากนัก อาจจะต้องค่อยๆ Render ไปทีละ Zoom Level

5.3 ข้อเสนอแนะ

1) Style Sheet ที่ทำไว้ ยังคงต้องทำการปรับปรุงเพื่อแสดงผลให้เหมือนกับที่ใช้งานอยู่บนเว็บไซต์
2) การอัพเดทฐานข้อมูล ต้องคอยตรวจสอบความถูกต้องทุกครั้ง
3) Style Sheet ที่มีการแชร์อยู่บนเว็บไซต์ไม่ค่อยได้รับการอัพเดท ควรมีการแจ้งให้ผู้พัฒนารับทราบ

เอกสารอ้างอิง

นักแผนที่. (2555). *GDAL/OGR*. (Online). Available: http://thaimapper.blogspot.com/2012/10/gdalgor.html. 16 พฤษภาคม 2559

วิกิพีเดีย. (2558). *ภาษาไพทอน*. (Online). Available: https://en.wikipedia.org/wiki/Python_%28programming_language%29. 16 พฤษภาคม 2559

เกี่ยวกับ PostgreSQL. (Online). Available: http://admin.wikidot.com/pgsqlintro. 16 พฤษภาคม 2559

GitHub. *mapnik/mapnik*. (Online). Available: https://github.com/mapnik/mapnik. 16 พฤษภาคม 2559

Mapnik - OpenStreetMap Wiki. (Online). Available: http://wiki.openstreetmap.org/wiki/Mapnik. 16 พฤษภาคม 2559

Open Source กับงานด้านสารสนเทศภูมิศาสตร์ (GIS). (Online). Available: http://kmcenter.rid.go.th/kcitc/2011/index.php?option=com_content&view=article&id=430:open-source-gis&catid=51:2011-08-25-08-19-47&Itemid=34. 16 พฤษภาคม 2559

Prajuab Riabroy. (2555). *ขั้นตอนการติดตั้ง PostGIS/Postgresql บน Ubuntu Server*. (Online). Available: https://priabroy.com/tag/postgis/. 16 พฤษภาคม 2559

ประวัติผู้จัดทำโครงงาน

ชื่อ – สกุล พงศกร อยู่สุข

วัน เดือน ปีเกิด 10 พฤศจิกายน 2537

ประวัติการศึกษา

ระดับประถมศึกษา ประถมศึกษาตอนปลายพ.ศ. 2544
โรงเรียนศรีวิกรม์

ระดับมัธยมศึกษา มัธยมศึกษาตอนปลาย พ.ศ. 2552
โรงเรียนเตรียมอุดมศึกษาพัฒนาการ

ระดับอุดมศึกษา คณะเทคโนโลยีสารสนเทศ สาขา เทคโนโลยีสารสนเทศ พ.ศ.2556
สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ทุนการศึกษา ทุนการศึกษา ประเภทที่ 1 สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ประวัติการฝึกอบรม

1. อบรมการทำ Cable ใต้น้ำ
2. ฟังบรรยายพิเศษ Space Technology สถาบันเทคโนโลยีไทย-ญี่ปุ่น
3.เข้าร่วมโครงการ CCNA NetRiders ปีพ.ศ. 2557

ผลงานที่ได้รับการตีพิมพ์

1. Design of Framework for Hospital Information System, ICBIR 2016
2. Design of Framework for Cross-Platform Application Development by using Phone Gap, ICBIR 2016