# บทนำ

ข้อมูลที่เราเก็บไว้ใน Computer ปัจจุบันมีข้อมูลที่เป็นชนิด Graph อยู่มากไม่ว่าจะเป็นหน้าเว็บเพจต่างๆ ที่เชื่อมโยงถึงกัน หรือไม่ว่าจะเป็น Diagram, Flowchart, Database, Network ล้วนเป็นข้อมูลที่เป็นชนิด graph ทั้งสิ้น หรือว่าจะเป็นข้อมูลของสาขาวิชาอื่นๆ ที่ไม่ใช่สาขาคอมพิวเตอร์โดยตรง เช่น ข้อมูลทางสถิติ พันธะเคมี สายเกรียว DNA ซึ่งข้อมูลเหล่านี้ส่วนใหญ่จะมีขนาดที่ใหญ่ และยากที่จะสามารถวิเคราะห์รูปแบบโครงแล้วเข้าใจได้ง่ายเมื่อข้อมูลเหล่าถูกแสดงอยู่ในรูปของ ตารางหรือในรูปแบบของข้อความ (text mode) เราจึงได้มีการนำข้อมูลเหล่านั้นมาปรับปรุงการแสดงผลให้อยู่ในรูปแบบของกราฟิก (graphic mode) เพื่อให้คนที่ดูข้อมูลเหล่านี้สามารถดูได้ง่ายขึ้น

โปรแกรมแสดงผลข้อมูลชนิดกราฟในรูปแบบกราฟิกนี้มีความสามารถในการจัดการกับกราฟ เช่นการ เพิ่ม,ลบ ,บันทึกกราฟ และ การเพิ่ม อัลกอลิทึม ที่จะจัดการกับกราฟแล้วนำมาแสดงผลได้ โดยไฟล์ที่ใช้กับโปรแกรมนี้เป็นชนิด GraphML ซึ่งเป็น xml ชนิดหนึ่งที่สามารถนำไปใช้กับโปรแกรมอื่นได้ และสามารถอ่านไฟล์จากโปรแกรมอื่นที่ใช้มาตรฐานเดียวกันได้ และข้อมูลจากโปรแกรมอื่นที่สามารถส่งออกข้อมูลเป็นประเภทนี้ ก็สามารถนำมาใช้กับโปรแกรมนี้เช่นกัน

โปรแกรมแสดงผลข้อมูลชนิดกราฟนี้เป็นโปรแกรมที่พัฒนาขึ้นโดยใช้ ภาษา Python และไลบรารีกราฟิกของ Python ที่ชื่อว่า PyQt4 ซึ่งทั้ง และตัวไลบราลีนี้ สามารถทำงานได้บนทุกระบบปฏิบัติการ (Platform Independent) ซึ่งทำให้ นำไปติดตั้ง และใช้กับระบบปฏิบัติการ ต่างๆได้ อีกทั้งโปรแกรมนี้ยังเป็น โอเพนซอร์ส เปิดโอกาสให้ทุกคนสามารถ ดาวน์โหลดไปใช้งานได้ฟรีหรือแม้กระทั่งจะพัฒนาต่อเพื่อนำไปใช้กับงานด้านอื่นอีกด้วย

การวิเคราะห์ข้อมูลประเภทกราฟในส่วนของโครงสร้างเป็นส่วนหนึ่งของการทำงานวิจัยที่มีขนาดใหญ่ การที่เราสารารถมองจุดเด่นต่างๆ ภายในกราฟออกมาได้จะมีส่วนช่วยทำให้เราสามารถคิดและวิเคราะห์ เพื่อสร้างสรรค์งานอื่นๆต่อไปได้ง่ายขึ้น หรือมีแนวทางที่จะพัฒนาไปในรูปแบบอื่นได้

## ที่มาของโครงงาน

ข้อมูลในสมัยนี้ส่วนมากอยู่ในรูปแบบของข้อมูลที่เชื่อมโยงกับข้อมูลอื่นเสมอ ซึ่งข้อมูลเหล่านี้มีขนาดใหญ่มากเมื่อเราต้องการจะนำข้อมูลเรานี้มาใช้เรามักจะไม่นำข้อมูลเหล่านี้มาใช้ตรงๆ(Data) แต่ว่าเราจะนำสารสนเทศที่ได้จะข้อมูลเหล่านี้มาใช้นั่นเอง (Information) ซึ่งรูปแบบหนึ่งของสารสนเทศ ที่ทำให้เห็นความสัมพันธ์ของข้อมูลเหล่านี้ได้ชัดเจนและเข้าใจได้ง่ายรูปแบบหนึ่งก็คือ การนำเสนอข้อมูลที่อยู่ในรูปแบบของกราฟ ออกมาในรูปกราฟิก

ข้อมูลที่อยู่ในรูปแบบกราฟปัจจุบันมีมากมายเช่น ไดอะแกรมการออกแบบระบบต่างๆ ต้นไม้ของระบบแฟ้มเอกสาร(Directory or Folder) ฐานข้อมูล(Database system) ระบบแท็ก(Tags or categories) หรือแม้แต่เว็บเพจที่เชื่อมโยงกันของแต่ล่ะเว็บไซต์ รวมไปถึงสิ่งที่เป็นฮาร์ดแวร์ เช่นสายไฟฟ้าที่เดินไฟฟ้ามาให้บ้านเรา การรู้จักกันของหมู่เพื่อน(Social Network) การเชื่อมโยงกันของระบบของเครือข่าย(Network system) ถนนที่เชื่อมเมืองต่างๆ เข้าด้วยกัน สิ่งเล็กๆอย่างแผงวงจร หรือจนถึงระดับนาโนอย่าง DNA ของสิ่งมีชีวิตหรือพันธะของสารเคมี ที่กล่าวมาข้างต้นทั้งหมดล้วนเป็นตัวอย่างของข้อมูลประเภทกราฟทั้งสิ้นแต่ว่าข้อมูลเหล่านี้บางอย่างอยู่ในรูปแบบที่เราไม่สามารถเข้าใจได้ง่ายเช่น การรู้จักกันของหมู่เพื่อนนั้นอาจเป็นสิ่งที่มีรูปแบบอะไรซ่อนอยู่แต่ว่าการที่เราจะทำความเข้าใจมันนั้นค่อนข้างยากซึ่งเราอาจจะต้องใช้การวิเคราะห์ข้อมูลหลายรูปแบบเข้าช่วยกันไม่ว่าจะเป็นทั้งรูปแบบของตารางความสัมพันธ์ หรือว่าในรูปแบบของกราฟิก

ด้วยเหตุผลข้างต้นผู้จัดทำโครงการจึงได้จัดทำโปรแกรมแสดงผลกราฟในรูปแบบของกราฟิกขึ้นมาเพื่อให้เป็นทางเลือกหนึ่งที่จะเป็นตัวช่วยสำหรับการวิเคราะห์ข้อมูลประเภทกราฟ

## แนวทางการพัฒนาโครงงาน

แนวทางในการพัฒนา และจัดทำโปรแกรมแสดงข้อมูลชนิดกราฟออกมาในรูปแบบกราฟิก มีดังต่อไปนี้คือ

1. ศึกษาเรื่องที่เกี่ยวข้อง
   1. กราฟและอัลกอริทึมที่เกี่ยวข้อง
   2. การสร้างการแสดงผลแบบกราฟิก
   3. ความเป็นไปได้ของเครื่องมือทีจะใช้ในการพัฒนา
   4. ประเภทไฟล์ที่ใช้ในการจัดเก็บข้อมูลประเภทกราฟ
2. กำหนดขอบเขตการทำงานให้เหมาะสมกับระยะเวลาในการดำเนินโครงการและความเป็นไปได้
3. ออกแบบโครงสร้างโปรแกรมรวมไปถึง รูปแบบการพัฒนา, กำหนดเครื่องมือ, ไฟล์ที่ใช้
4. พัฒนาโปรแกรม
5. ทดสอบการทำงานของโปรแกรม
6. จัดทำเอกสารประกอบการดำเนินงาน

## วัตถุประสงค์

โปรแกรมแสดงผลข้อมูลประเภทกราฟนี้จัดทำขึ้น เพื่อที่จะช่วยในงานด้านต่างๆเกี่ยวกับกราฟดังนี้คือ

1. การแสดงผลในข้อมูลชนิดกราฟออกมาในรูปแบบกราฟิก ทำให้มองภาพรวมของกราฟออกมาได้ในอีกรูปแบบหนึ่งแทนที่จะเป็นตารางหรือข้อความ
2. แสดงผลการทำงานของอัลกอริทึมให้ออกมาเป็นรูปธรรมสามารถมองเห็นได้ชัด เพื่อทดสอบการทำงานของอัลกอริทึมต่างๆ
3. สร้างไฟล์กราฟที่สามารถนำไปใช้กับงานด้านอื่นต่อได้
4. นำข้อมูลที่เป็นชนิดกราฟที่ได้จากโปรแกรมหรือข้อมูลจากที่อื่นมาวิเคราะห์ได้

## ขอบเขตของโครงงาน

โปรแกรมแสดงผลข้อมูลประเภทกราฟนี้จัดทำขึ้นเพื่อให้สามารถอ่านข้อมูลที่เป็นชนิดกราฟขึ้นมาแสดงผลในรูปแบบของกราฟิกได้ และต้องมีส่วนของการจัดการกราฟต่างๆ ดังนี้

1. การแสดงผลในข้อมูลชนิดกราฟออกมาในรูปแบบกราฟิกโดยที่สามารถจัดการกับกราฟในกราฟิกได้ในระดับหนึ่งเช่น การจัดการกราฟด้วยเมาส์เพื่อเปลี่ยนพิกัดของจุดตัด
2. สารารถดัดแปลงแก้ไขกราฟนั้นๆได้เช่นสามารถเพิ่มเส้นเชื่อมหรือจุดตัดได้ในขณะที่โปรแกรมทำงานอยู่
3. สามารถบันทึกกราฟหลังการแก้ไขในข้อ 2 ได้โดยรูปแบบของการจัดเก็บนั้นต้องเป็นสากลและสามารถนำไปเปิดกับโปรแกรมอื่นๆได้
4. การอ่านไฟล์ที่ได้จากการบันทึกในข้อ 3 ได้และสามารถอ่านไฟล์มาตรฐานเดียวกันที่บันทึกจากโปรแกรมอื่นได้
5. การจัดการกับกราฟด้วยอัลกอริทึม ที่จะให้กราฟแสดงข้อมูลที่ได้จากการประมวลผลด้วยอัลกอริทึม

## ทรัพยากรที่ใช้ในการจัดทำโครงงาน

ทรัพยากรที่ใช้ในการจัดทำโปรแกรมแสดงผลกราฟในรูปแบบของกราฟิกซึ่งประกอบด้วย อุปกรณ์ด้านฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบแบ่งได้ดังนี้ คือ

1. ด้านฮาร์ดแวร์(Hardware)
   * เครื่องคอมพิวเตอร์ส่วนบุคคล(Notebook, PC, Macintosh ,MacBook)
   * ซีพียู 1.60 GHz
   * แรม 256 MB
   * ฮาร์ดดิสก์ความจุ 4 GB
2. ด้านซอฟแวร์(Software)
   * Python runtime : ภาษาที่ใช้ในการพัฒนา
   * PyQt4 library : library ของ Python ที่ใช้ในการสร้างกราฟิก
   * Easy Eclipse for LAMP : IDE ที่ใช้ในการพัฒนา
   * yEd graph editor : โปรแกรมในการสร้างกราฟตัวอย่าง
3. Server control version
   * Google Code: SVN repository

## ระยะเวลาในการดำเนินงาน

ระยะเวลาที่ใช้ในการพัฒนาโปรแกรม ดังตารางที่ 1-1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **การดำเนินงาน** | **มิ.ย.** | **ก.ค.** | | **ส.ค.** | | **ก.ย.** | | | **ต.ค.** | |
| 1. ศึกษาปัญหา และแนวทางการแก้ปัญหา |  |  | |  | |  | | |  | |
| 2. วิเคราะห์ และออกแบบโปรแกรม |  |  |  |  | |  |  | |  | |
| 3. เขียนโปรแกรม |  |  | |  | |  | | |  |  |
| 4. ทดสอบการทำงานของโปรแกรม |  |  | |  |  |  | | |  |  |
| 5. จัดทำเอกสาร |  |  | |  | |  | |  |  |  |

**ตารางที่ 1-1** ระยะเวลาในการดำเนินงาน

## ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการทำโปรแกรมแสดงผลข้อมูลชนิดกราฟในรูปแบบกราฟิกมีดังนี้

1. ช่วยให้การทำความเข้าใจข้อมูลประเภทกราฟง่ายขึ้น
2. ทดสอบความถูกต้องของอัลกอริทึมที่ใช้วิเคราะห์กราฟได้
3. มีโปรแกรมแสดงผลข้อมูลชนิดกราฟให้อยู่ในรูปที่ดูง่ายขึ้น
4. สามารถนำกราฟที่สร้างขึ้นไปใช้ต่อได้

# ทฤษฏีที่เกี่ยวข้อง

ทฤษฏีและกระบวนการต่างที่เกี่ยวข้องในการจัดทำโปรแกรมแสดงผลข้อมูลชนิดกราฟออกมาในรูปแบบกราฟิก มีดังนี้

2.1 การควบคุมเวอร์ชันของรหัสโปรแกรมด้วย (SVN **S**ub**v**ersio**n)** และ Google Code

2.2 XML และ GraphML

2.3 Python และ PyQt4

2.4 กราฟอัลกอริทึม

2.5 IDE, Eclipse และ EasyEclipse for LAMP

2.6 Open Source Software

ซึ่งจะอธิบายถึงรายละเอียด วิธีใช้ ตัวอย่าง และการนำมาใช้ของแต่ล่ะหัวข้อข้างต้นโดยละเอียดต่อไปนี้

## การควบเวอร์ชันของรหัสโปรแกรมด้วย (SVN Subversion) และ Google Code

เนื่องจากขณะที่โปรแกรมยังอยู่ในขั้นตอนของการพัฒนานั้นจะมีการเปลี่ยนแปลงของโปรแกรมอย่างมากและอาจจะมีการเปลี่ยนโครงสร้างของโปรแกรมด้วยซึ่งการแก้ไขบางอย่างอาจจะทำให้โปรแกรมเสียหายหรือไม่สามารถใช้งานได้ เราอาจแก้ปัญหานี้ได้โดยการทำ สำรองข้อมูลไฟล์(back up file) สำหรับแต่ล่ะรุ่นของโปรแกรม แต่ว่าการทำสำรองข้อมูลทั้งหมดของโปรแกรม ทุกๆเวอร์ชันจะทำให้เปลืองพื้นที่ในฮาร์ดดิสก์มาก ซึ่งปัจจุบันปัญหาแบบนี้ได้มีผู้ที่คิดวิธีในการแก้ไขปัญหาแบบนี้แล้วคือการใช้ ซอฟแวร์ในการควบคุมรหัสโปรแกรม(source code) โดยโปรแกรมที่เป็นที่รู้จักกันมากในปัจจุบันได้แก่ SVN (**S**ub**v**ersio**n**)และ CVS (Concurrent Versions System) ซึ่งในที่นี้ผู้พัฒนาได้เลือกใช้ SVN เพราะว่า SVN สนับสนุนการ เปลี่ยนชื่อไฟล์ และการลบไดเรคทอรี ซึ่งเป็นเรื่องสำคัญในช่วงแรกของการพัฒนาโปรแกรมที่การออกแบบยังไม่สมบูรณ์ และเลือก Server ที่ใช้ในการเก็บที่เป็นที่น่าเชื่อถืออย่าง Google code

SVN หรือ Subversion เป็นโปรแกรมโอเพนซอร์สที่ใช้ในการควบคุมรหัสโปรแกรม ที่ริเริ่มจัดทำโดยบริษัท CollabNet Inc. โดย SVN นี้พัฒนาต่อมาจากโปรแกรม CVS ที่มีมาก่อนหน้า และเพิ่มความสามารถต่างๆ เข้ามาจนปัจจุบันมีผู้ใช้เยอะขึ้นมาก และมีโปรแกรมที่สนับสนุนในการใช้มาก เช่น plug-in ของโปรแกรม Eclipse ทำให้การใช้ SVN ไม่ได้ยึดติดกับการใช้ส่วนติดต่อผู้ใช้แบบ Text mode เท่านั้น แต่ว่าสามารถใช้ GUI ต่างๆ เข้ามาช่วยได้

Google Code เป็นบริการแบบหนึ่งของ Google ที่มีไว้เพื่อสนับสนุนการทำงานของโปรแกรมเมอร์โอเพนซอร์ส ซึ่งมีบริการในการจัดการเกี่ยวกับโครงการที่นักพัฒนาต้องการสร้างอยู่หลายอย่างเช่น

* Wiki สำหรับอภิปรายเกี่ยวกับโครงการในกรณีที่ทำร่วมกันหลายคน หรือว่าเอาไว้ทำเอกสารออนไลน์เกี่ยวกับโครงการได้
* Download เป็นที่สำหรับเก็บเอกสาร โปรแกรมหรือสิ่งที่จำเป็นต่างสำหรับโครงการและกำหนดได้ว่าใครสามารถดาวน์โหลด ซึ่งอาจจะเป็นกลุ่มผู้พัฒนาหรือบุคคลทั่วไปก็ได้
* Issue การตามรุ่นต่างๆของโปรแกรมเช่นในแต่ละรุ่นมีการพัฒนามีการแก้ไขหรือเปลี่ยนแปลอะไรบ้าง
* Source เป็นที่สำหรับบริการให้โหลดรหัสโปรแกรมซึ่งมีสองแบบคือ
  + กลุ่มผู้พัฒนา(มีการส่งซอร์สโค้ด ที่ได้ทำการแก้ไขกลับไปที่เซิร์ฟเวอร์ )
  + บุคคลทั่วไป(ไม่มีการส่งซอร์สโค้ดที่ได้ทำการแก้ไขกลับไปที่เซิร์ฟเวอร์ )

ซึ่งยังมีการให้บริการในการควบคุมรุ่นของโปรแกรมด้วย svn ด้วยซึ่งผู้พัฒนาได้ใช้เซิร์ฟเวอร์  ของ Google นี้เป็นเซิร์ฟเวอร์  svn สำหรับการพัฒนาโปรแกรมแสดงผลข้อมูลประเภทกราฟออกมาในรูปแบบกราฟิกด้วย

## Python

* **ข้อมูลเบื้องต้น**

ไพธอน (Python) เป็นภาษาโปรแกรมในลักษณะภาษาอินเตอร์พรีเตอร์โปรแกรมมิ่ง (Interpreted programming language) ผู้คิดค้นคือ Guido van Rossum ในปี 1990 ซึ่ง Python เป็น การจัดการชนิดของตัวแปรแบบแปรผันตามข้อมูลที่บรรจุอยู่ (Fully dynamically typed) และใช้การจัดการหน่วยความจำเป็นอัตโนมัติ (Automatic memory management) โดยได้เป็นการพัฒนาและผสมผสานของภาษาอื่น ๆ ได้แก่ ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl และภาษา Python ยังเป็นแนวคิดที่ทำให้เกิดภาษาใหม่ ๆ ซึ่งได้แก่ Ruby และ Boo เป็นต้น

Python นั้นพัฒนาเป็นโครงการ Open source โดยมีการจัดการแบบไม่หวังผลกำไรโดย Python Software Foundation และสามารถหาข้อมูลและตัวแปรภาษาได้จากเว็บไซต์ของ Python เอง ที่ http://www.python.org/ ซึ่งในปัจจุบัน (ณ.วันที่ 29 กันยายน 2551) Python ได้พัฒนาถึงรุ่นที่ 2.5.2 (ออกวันที่ 22 กุมภาพันธ์ 2551) และรุ่นทดสอบการทำงาน หรือ beta นั้นอยู่ที่รุ่น 2.6rc2(12 กันยายน 2551) และ 3.0rc1 (17 กันยายน 2551)

* **Python 1.0**

Python สร้างขึ้นครั้งแรกในปี 1990 โดย Guido van Rossum ที่ CWI (National Research Institute for Mathematics and Computer Science) ในประเทศเนเธอร์แลนด์ โดยได้นำความสำเร็จของภาษาโปรแกรมมิ่งที่ชื่อ ABC มาปรับใช้กับ Modula-3, Icon, C, Perl, Lisp, Smalltalk และ Tcl โดย Duido van Rossim ถือว่าเป็นผู้ริเริ่มและคิดค้น แต่เค้าก็ยังคิดว่าผลงานอย่าง Python นั้น เป็นผลงานความรู้ที่ทำขึ้นเพื่อความสนุกสนานโดยได้อ้างอิงงานชิ้นนี้ของเขาว่าเป็น Benevolent Dictator for Life (BDFL) ซึ่งผลงานที่ถูกเรียกว่าเกิดจากความสนุกสนานเหล่านี้นั้นมักถูกเรียกว่า BDFL เพราะมักเกิดจากความไม่ตั้งใจ และความอยากที่จะทำอะไรที่เป็นอิสระนั้นเอง ซึ่งคนที่ถูกกล่าวถึงว่าทำในลักษณะแบบนี้ก็ได้แก่ Linus Torvalds ผู้สร้าง Linux kernel, Larry Wall ผู้สร้าง Perl programming language และคนอื่น ๆ อีกมากมาย

โดยที่ใน Python 1.2 นั้นได้ถูกปล่อยออกมาในปี 1995 โดย Guido ได้กลับมาพัฒนา Python ต่อที่ Corporation for National Research Initiatives (CNRI) ที่ เรสตัน, มลรัฐเวอร์จิเนีย ประเทศสหรัฐอเมริกา โดยที่ในขณะเดียวกันก็ได้ปล่อยรุ่นใหม่ ในหมายเลขรุ่น 1.6 ออกมาโดยอยู่ที่ CNRI เช่นกัน

ซึ่งหลังจากปล่อยรุ่น 1.6 ออกมาแล้ว Guido van Rossum ก็ได้ออกจาก CNRI  เพื่อทำงานให้การทำธุรกิจพัฒนาซอฟต์แวร์แบบเต็มตัว โดยก่อนที่จะเริ่มทำงานธุรกิจ เขาก็ได้ทำให้ Python นั้นอยู่บนสัญญาลิขสิทธิ์แบบ General Public License (GPL) โดยที่ CNRI และ Free Software Foundation (FSF) ได้รวมกันเปิดเผยรหัสโปรแกรมทั้งหมด เพื่อให้ Python นั้นได้ชื่อว่าเป็นซอฟต์แวร์เสรี และเพื่อให้ตรงตามข้อกำหนดของ GPL-compatible ด้วย (แต่ยังคงไม่สมบูรณ์เพราะการพัฒนาในรุ่น 1.6 นั้นออกมาก่อนที่จะใช้สัญญาลิขสิทธิ์แบบ GPL ทำให้ยังมีบางส่วนที่ยังเปิดเผยไม่ได้)

และในปีเดียวกันนั้นเอง Guido van Russom ก็ได้รับรางวัลจาก FSF ในชื่อว่า "Advancement of Free Software"

โดยในปีนั้นเอง Python 1.6.1 ก็ได้ออกมาเพื่อแก้ปัญหาข้อผิดพลาดของตัวซอฟต์แวร์และให้เป็นไปตามข้อกำหนดของ GPL-compatible license อย่างสมบูรณ์

* **Python 2.0**

ในปี 2000 Guido และ Python Core Development team ได้ย้ายการทำงานไป BeOpen.com โดยที่พวกเขาได้ย้ายจาก BeOpen PythonLabs team โดยใน Python รุ่นที่ 2.0 นั้นได้ถูกนำออกเผยแพร่ต่อบุคคลทั่วไปจากเว็บไซต์ BeOpen.com และหลังจากที่ Python ออกรุ่นที่ 2.0 ที่ BeOpen.com แล้ว Guido และนักพัฒนาคนอื่น ๆ ในทีม PythonLabs ก็ได้เข้ารวมกับทีมงาน Digital Creations

Python รุ่น 2.1 ได้สืบทอนการทำงานและพัฒนามาจาก 1.6.1 มากกว่า Python รุ่น 2.0 และได้ทำการเปลี่ยนชื่อสัญญาลิขสิทธิ์ใหม่เป็น Python Software Foundation License โดยที่ใน Python รุ่น 2.1 alpha นั้นก็ได้เริ่มชื่อสัญญาสิขสิทธิ์นี้และผู้เป็นเจ้าของคือ Python Software Foundation (PSF) โดยที่เป็นองค์กรที่ไม่หวังผลกำไรเช่นเดียวกับ Apache Software Foundation

* **อนาคต**

ผู้พัฒนา Python มีการประชุมและถกเถียงกันในเรื่องของความสามารถใหม่ ๆ ใน Python รุ่นที่ 3.0 โดยมีชื่อโครงการว่า Python 3000 (Py3K) โดยที่จะหยุดการสนับสนุนโค้ดโปรแกรมจากรุ่น 2.x โดยที่ทำแบบนี้เพื่อทำการปรับปรุงเปลี่ยนแปลงการทำงานของภาษาให้ดียิ่งขึ้นตามคำแนะนำที่ว่า "reduce feature duplication by removing old ways of doing things" (ลดทอนคุณสมบัติที่ซ้ำซ้อนด้วยการยกเลิกเส้นทางที่เดินผ่านมาแล้ว) โดยในตอนนนี้ยังไม่มีตารางงานของ Python รุ่น 3.0 แต่อย่างใด แต่ Python Enhancement Proposal (PEP) ได้มีการวางแผนไว้แล้ว โดยได้วางแผนไว้ดังนี้

* + ทำการเพื่อส่วนสนับสนุนชนิดตัวแปรให้มากขึ้น
  + สนับสนุนการทำงานของชนิดตัวแปรแบบ unicode/str และ separate mutable bytes type
  + ยกเลิกการสนับสนุนคุณสมบัติของ classic class, classic division, string exceptions และ implicit relative imports
  + ฯลฯ
* **หลักปรัชญาของภาษาไพธอน**

Python เป็นภาษาที่สามารถสร้างงานได้หลากหลายกระบวนทัศน์ (Multi-paradigm language) โดยจะมองอะไรที่มากกว่าการ coding เพื่อนำมาใช้งานตามรูปแบบเดิม ๆ แต่จะเป็นการนำเอาหลักการของกระบวนทัศน์ (Paradigm) แบบ Object-oriented programming, Structured programming, Functional programming และ Aspect-oriented programming นำเอามาใช้ทั้งแบบเดียว ๆ และนำมาใช้ร่วมกัน ซึ่ง Python นั้นเป็น ภาษาที่มีการตรวจสอบชนิดตัวแปรแบบยืดหยุ่น (dynamically type-checked) และใช้ Garbage collection ในการจัดการหน่วยความจำ

* **ข้อเด่นของภาษาไพธอน**
* ง่ายต่อการเรียนรู้ โดยภาษา Python มีโครงสร้างของภาษาไม่ซับซ้อนเข้าใจง่าย ซึ่งโครงสร้างภาษา Python จะคล้ายกับภาษาซีมาก เพราะภาษา Python สร้างขึ้นมาโดยใช้ภาษาซี ทำให้ผู้ที่คุ้นเคยภาษาซี อยู่แล้วใช้งานภาษา Python ได้ไม่ยาก นอกจากนี้โดยตัวภาษาเองมีความยืดหยุ่นสูงทำให้การจัดการกับงานด้านข้อความ และ Text File ได้เป็นอย่างดี
* ไม่ต้องเสียค่าใช้จ่ายใดๆ ทั้งสิ้น เพราะตัวแปรภาษา Python อยู่ภายใต้ลิขสิทธิ์ Python Software Foundation License (PSFL) ซึ่งเป็นของ Python Software Foundation (PSF) ซึ่งมีลักษณะคล้ายกับ ลิขสิทธิ์แม่แบบอย่าง General Public License (GPL) ของ Free Software Foundation (FSF)
* ใช้ได้หลายแพลตฟอร์ม ในช่วงแรกภาษา Python ถูกออกแบบใช้งานกับระบบ Unix อยู่ก็จริง แต่ในปัจจุบันได้มีการพัฒนาตัวแปลภาษา Python ให้สามารถใช้กับระบบปฏิบัติการอื่นๆ อาทิเช่น Linux Platform, Windows Platform, OS/2, Amiga, Mac OS X และรวมไปถึงระบบปฎิบัติการที่ .NET Framework, Java virtual machine ทำงานได้ ซึ่งใน Nokia Series 60 ก็สามารถทำงานได้เช่นกัน
* ภาษา Python ถูกสร้างขึ้นโดยได้รวบรวมเอาส่วนดีของภาษาต่างๆ เข้ามาไว้ด้วยกัน อาทิเช่น ภาษา ABC, Modula-3, Icon, ANSI C, Perl, Lisp, Smalltalk และ Tcl
* Python สามารถรวมการพัฒนาของระบบเข้ากับ COM, .NETและ CORBA objects
* สำหรับ Java libraries แล้วสามารถใช้ Jython เพื่อทำการพัฒนาซอฟต์แวร์จากภาษา Python สำหรับ Java Virtual Machine
* สำหรับ .NET Platform แล้ว สามารถใช้ IronPython ซึ่งเป็นการพัฒนาของ Microsoft เพื่อจะทำให้ Python นั้นสามารถทำงานได้บน .Net Framework ซึ่งใช้ชื่อว่า Python for .NET
* Python นั้นสนับสนุน Internet Communications Engine (ICE) และการรวมกันของเทคโนโลยีอื่น ๆ อีกมากมายในอนาคต
* บางครั้งนักพัฒนาอาจจะพบว่า Python ไม่สามารถทำงานบางอย่างได้ แต่นักพัฒนาต้องการให้มันทำงานได้ ก็สามารถพัฒนาเพิ่มได้ในรูปแบบของ extension modules ซึ่งอยู่ในรูปแบบของโค้ด C หรือ C++ หรือใช้ SWIG หรือ Broost.Python
* ภาษา Python เป็นสามารถพัฒนาเป็นภาษาประเภท Server side Script คือการทำงานของภาษา Python จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมายัง Client ทำให้มีความปลอดภัยสูง และยังใช้ภาษา Python นำมาพัฒนาเว็บเซอร์วิสได้อีกด้วย
* ใช้พัฒนาระบบบริหารการสร้างเว็บไซต์สำเร็จรูปที่เรียกว่า Content Management Systems (CMS) ซึ่ง CMS ที่มีชื่อเสียงมาก และเบื้องหลังทำงานด้วย Python คือ Plone http://www.plone.org/
* **Category และ Application Domains**

ภาษา Python นั้น จัดอยู่ใน Category ภาษาที่สามารถสร้างงานได้หลากหลายกระบวนทัศน์ (Multi-paradigm language) โดยรองรับทั้ง Object-oriented programming, Imperative, Functional programming และ Logic programming ซึ่ง Python สามารถนำไปพัฒนาซอฟต์แวร์ประยุกต์ได้มากมาย ได้แก่

* **Web และ Internet Development**

Python นั้นมีการสนับสนุนในด้านของ Web Development ในโซลูชันระดับสูงด้วย Zope, mega frameworks อย่าง Django และ TurboGears และรวมไปถึง Content Management Systems ขั้นสูงอย่าง Plone และ CPS จึงทำให้ Python นั้นเป็น Common Gateway Interface (CGI) ระดับสูงที่มีประสิทธิภาพที่ดีที่สุดตัวหนึ่งในตลาด

* **Database Access**

Python นั้นสนับสนุนการเข้าถึงข้อมูลในฐานข้อมูลของผู้ผลิตฐานข้อมูลต่าง ๆ มากมาย โดยผ่านทาง ODBC Interfaces และ Database Connection Interface อื่น ๆ ซึ่งสามารถทำงานร่วมกับ  MySQL, Oracle, MS SQL Server, PostgreSQL, SybODBC และอื่น ๆ ที่จะมีมาเพิ่มเติมอีกในอนาคต

* **Desktop GUIs**

เมื่อ Python ได้ติดตั้งลงบนเครื่องของคุณแล้ว จะมี Tk GUI development library ซึ่งเป็น libraries ที่มีความสามารถเทียบเท่า Microsoft Foundation Classes (MFC, ซึ่งคล้าย ๆ กับ win32 extensions), wxWidgets, GTK, Qt, Delphi และอื่น ๆ ทำให้สามารถพัฒนาซอฟต์แวร์ประยุกต์ต่าง ๆ แบบ Graphic user interface ได้

* **Scientific และ Numeric computation**

Python รองรับการทำงานของนักวิทยาศาสตร์ในเรื่องของทฤษฎีการคำนวณ, Bioinformatics และ Physics เป็นต้น

* **Education**

Python นั้นเป็นภาษาที่เหมาะกับการเรียนการสอนในวิชา programming อย่างมาก โดยสามารถนำไปใช้ในระดับเบื้องต้นถึงระดับสูง ซึ่ง Python Software Foundation นั้นได้มีหลักสูตรสำหรับการเรียนการสอนในด้านนี้อยู่แล้ว ซึ่งสามารถนำเอา pyBiblio  และ Software Carpentry Course มาเรียนเพื่อเสริมความรู้ได้

* **Network programming**

เป็นการเพิ่มความสามารถจาก Web และ Internet Development Python นั้นสนับสนุนในการเขียนโปรแกรมในระดับต่ำในด้านของ network programming ที่ง่ายต่อการพัฒนา sockets และ รวมไปถึงการทำงานร่วมกับ modules อย่าง Twisted และ Framework สำหรับ Asynchronous network programming

* **Software build และ Testing**

Python นั้นสนับสนุนการพัฒนาซอฟต์แวร์ที่มีการควบคุมการพัฒนาและจัดการระบบทดสอบต่าง ๆ โดยใช้เครื่องมือในการพัฒนาที่สนับสนุนการเขียนโปรแกรมใน Python เอง ซึ่งตัว Python นั้นได้มาพร้อมกับ

* Scons สำหรับ build โปรแกรม
* Buildbot และ Apache Gump ที่ใช้สำหรับงาน Automated continuous compilation และ Testing
* Roundup หรือ Track สำหรับ bug tracking และ project management
* **Game และ 3D Graphics Rendering**

Python นั้นได้ถูกใช้ในตลาดพัฒนาเกมส์ทั้งเชิงธุรกิจและสมัครเล่น โดยมีการสร้าง Framework สำหรับพัฒนา Game บน Python ซึ่งชื่อว่า PyGame และ PyKyra ซึ่งยังรวมไปถึงการทำ 3D Graphics Rendering ที่ Python มี libraries ทางด้านงานนี้อยู่มากมาย

* **ซอฟต์แวร์ที่เขียนด้วยไพธอน**
* **BitTorrent**เป็นการพัฒนาโดยระบบการจัดการไฟล์ BitTorrent, การจัดการ การกระจายตัวของ Package ข้อมูลใน Tracker และการเข้ารหัสส่วนข้อมูลต่าง ๆ
* **Blender**ซอฟต์แวร์ open source สำหรับทำ 3D modeling
* **Chandler**ซอฟต์แวร์จัดการข้อมูลส่วนบุคคล (Personal Information Manager, PIM) โดยมีส่วนเพิ่มเติมทั้งงานปฎิทิน, อีเมล, ตารางงาน และข้อมูลโน็ตต่าง ๆ ซึ่งทำงานคล้าย ๆ กับ Outlook ของ Microsoft
* **Civilization IV** วีดิโอเกมส์ และยังเป็นเกมส์ที่ใช้ boost.python เพื่อทำการควบคุมส่วนประกอบต่าง ๆ ภายในเกมส์ ซึ่งรวมไปถึงรูปแบบ, หน้าตา และเนื้อหาของเกมส์ด้วย
* **Mailman**หนึ่งในซอฟต์แวร์ E-Mail mailing lists ที่ได้รับความนิยมสูงสุด
* **Kombilo**ระบบจัดการฐานข้อมูลของเกมส์โกะ
* **MoinMoin**ระบบ Wiki ที่ได้รับความนิยมสูงตัวหนึ่ง
* **OpenRPG**ระบบเกมส์เสมือนแบบ Role Playing Games ลน Internet
* **Plone**ระบบ Content Management System
* **Trac**ระบบติดตามติดตามข้อผิดพลาดและจัดการข้อมูลด้านการพัฒนาซอฟต์แวร์ด้วย MoinMoin ที่เป็น wiki และ Subversion เพื่อทำระบบ Source version control
* **Turbogears**ระบบพัฒนาซอฟต์แวร์ Framework โดยรวมเอา Cherrypy, SQLObject, MochiKit และ KID templates
* **ViewVC**ระบบ Web-based สำหรับจัดการด้าน CVS และ SVN repositories
* **Zope**ระบบพัฒนาซอฟต์แวร์บนอินเทอร์เน็ตแบบ web-application platform
* **Battlefield 2** เกมส์ First Person Shooter ที่ได้ใช้ Python ในการทำ Configuration scripts
* **Indian Ocean Tsunami Detector**ซอฟต์แวร์สำหรับมือถือเพื่อแจ้งเตือน Tsunami
* **EVE Online** เกมส์แบบ Multi Massive Online Role Playing Game ซึ่งเป็นเกมส์ที่ได้รับอันดับสูงมากบน MMORPG.com
* **SPE - Stani’s Python Editor**เป็น Free และ open-source สำหรับงานพัฒนาซอฟต์แวร์ด้วย Python โดยมีทั้งแบบ Python IDE for Windows, Linux \& Mac with wxGlade (GUI designer), PyChecker (Code Doctor) และ Blender (3D)

## Qt

Qt (อ่านว่า cute) เป็น cross-platform framework ในการพัฒนาโปรแกรมที่ถุกพัฒนาขึ้นโดย Trolltech เพื่อใช้ในการพัฒนาโปรแกรมในรูปแบบกราฟิกหรือเป็นที่รู้จักกันในอีกชื่อว่า widget-toolkit แล้วก็ยังถูกใช้ในพัฒนาโปรแกรมที่เป็นแบบ Text mode หรือว่าโปรแกรมที่ทำงานอยู่บนเซิร์ฟเวอร์ก็ได้ Qt ถูกใช้อย่างเห็นได้ชัดในหลายโปรแกรมเช่น KDE(Linux Desktop Environment) ,Opera (Web browser),Google Earth, Skype, Photoshop Element ,VirtualBox ,VLC Media Playerเป็นต้น

Qt สร้างขึ้นมาโดยใช้ C++ หลายส่วนขยายของโปรแกรมที่ไม่เป็นมาตรฐานถูกเขียนขึ้นโดยเพิ่มเข้ามากับมาโค้ดที่เป็นมาตรฐานของ C++ ก่อนที่จะ compile ออกมาเป็น Qt โดยเป้าหมายการการพัฒนาโปรแกรมของ Qt มีแนวความคิดมากจากคำว่า

*“Code less Create more Deploy everywhere”*

*“เขียนโค้ดน้อยลง สร้างสรรค์ผลงานได้มากขึ้น นำไปใช้ได้ทุกที่”*

ดังนั้น Qt แม้ว่าจะเขียนขึ้นมาด้วย C++ แต่ว่าก็สามารถที่นำไปใช้กับภาษาโปรแกรมภาษาอื่นได้ เช่น Ada (QtAda), C# (Qyoto/Kimono),Java(Qt Jambi), Pascal, Perl, PHP(PHP-Qt),Ruby( RubyQt) และภาษาที่ผู้พัฒนาได้เลือกมาใช้กับ framework นี้คือ Python(PyQt) ซึ่งทั้งตัวภาษาและ framework สามารถทำงานได้บน platforms หลักๆ ได้ทั้งหมดและยังสนับสนุนเรื่องภาษาท้องถิ่นจำนวนมากรวมถึงภาษาไทย มีความสามารถอื่นๆที่ไม่ใช่ด้านกราฟิกเพื่อสนับสนุนการเขียนโปรแกรมอีกได้แก่ SQL, XML, การจัดการกับการทำงานแบบ Thread ,Network ,และการจัดการไฟล์ข้าม platform

Qt ถูกพัฒนาออกมา 4 Editions สำหรับจุดประสงค์ต่างกันคือ

* Qt Console Edition สำหรับพัฒนาโปรแกรมแบบคอนโซลคือเป็น Text mode หรือโปรแกรมที่ทำงานเบื้องหลัง
* Qt Desktop Light สำหรับเขียนโปรแกรม GUI ที่มีขนาดเล็ก โดยตัดความสามารถเรื่อง network และ database ออกไป
* Qt Desktop Edition สมบูรณ์ที่มีทุกความสามารถครบ
* Qt Open Source Edition สมบูรณ์ที่มีข้อยกเว้นเกี่ยวกับเรื่องสัญญาอนุญาตของ ซอฟต์แวร์ (ซึ่งเป็นรุ่นที่ผู้พัฒนาเลือกนำมาใช้ในการพัฒนาโปรแกรม)

Qt ถูกพัฒนาขึ้นมาแล้วทั้งหมด 4 รุ่นซึ่งรุ่นล่าสุดคือรุ่นที่ 4 ซึ่งมีความสามารถหลักๆเพิ่มขึ้นมาอีก 4 อย่างคือ

* เซทของ Template
* Interviews เพิ่มหลักการของ model/views ให้กับวัตถุที่ใช้ในการแสดงผล(item views)
* Arthur framework สำหรับการสร้างรูปสองมิติ
* Scribe การ render ข้อความที่เข้ารหัสแบบ Unicode
* MainWindow ส่วนเสริมที่ในการสร้างหน้ากากหลักของโปรแกรม

ซึ่งในรุ่นที่ 4 ของ Qt ก็ได้ออกรุ่นย่อยออกมา 4 รุ่นซึ่งแต่ล่ะรุ่นก็มีความสามารถใหม่ๆเพิ่มขึ้นมาซึ่งผู้พัฒนาได้เลือกรุ่นล่าสุดมาใช้คือรุ่น 4.4 เนื่องจากก่อนหน้านี้ยังไม่สนับสนุน Vista และรุ่นนี้ก็เพิ่มความสามารถแล้วแก้ไขปัญหาของรุ่นก่อนหน้าเรียบร้อย

### PyQt4

เซทของ python แปลงมาจาก Qt4 เพื่อให้สามารถนำมาเขียนได้ง่ายในรูปแบบภาษาแบบ python ซึ่งได้รวมเอาความสามารถต่างๆ ของ Qt4 ไว้ครบถ้วนซึ่งผู้ใช้ไม่ต้องเรียนรู้ Syntax ของภาษาใหม่แต่อย่างใดแต่ต้องเรียนรู้วิธีการใช้ และหลักการที่เราจะนำมาใช้ในการพัฒนาโปรแกรมซึ่งทำให้ลดเวลาในการปรับตัวลงได้มาก และทำให้เราสามารถสร้างงานได้มากขึ้น

## Eclipse

Eclipse เป็นเครื่องมือที่ช่วยสำหรับในการพัฒนาโปรแกรมสำหรับนักพัฒนาโปรแกรม Java เป็นหลัก โดยตัวโปรแกรมเองเป็นซอฟต์แวร์โอเพนซอร์ส ซึ่งปัจจุบันได้มีคนจำนวนมากได้ช่วยกันพัฒนาโปรแกรมตัวนี้ให้สามารถพัฒนาโปรแกรมภาษาอื่นเพิ่มขึ้นได้อีกเป็นจำนวนมากและมีการสร้างเครื่องมือเสริมอื่นๆขึ้นมาอีกเป็นจำนวนมากเช่น การควบคุมรุ่นของโปรแกรมด้วย SVN, CVSเครื่องมือการเขียนโปรแกรมภาษาอื่นๆ Pydev(ภาษา Python ), RadRail (Ruby on rail), PHP, C/C++ เครื่องมือเกี่ยวกับการพัฒนาเว็บเช่น browser ภายในตัวเอง, XML Editor, Database tool เครื่องมือในการวัดประสิทธิภาพของโปรแกรม profiler, Enerjy

อย่างไรก็ตามเครื่องมือเสริมต่างๆ ข้างต้นถ้าเราต้องการใช้เราต้องติดตั้งเพิ่มขึ้นเอง แต่ก็มีกลุ่มของผู้พัฒนาที่ใช้งานบางอย่างเหมือนๆ กันเช่น กลุ่มของนักพัฒนาเว็บ กลุ่มของนักพัฒนาโปรแกรมด้วยภาษาต่างๆ กลุ่มที่พัฒนาโปรแกรมเฉพาะทาง จึงได้มีกลุ่มที่คิดที่จะทำ เครื่องมือเสริมของ Eclipse รวมกับ Eclipse ไว้เป็นชุดๆ เพื่อความง่ายในการเฉพาะทางโดนไม่ต้องมาเสียเวลาในการลงเครื่องมือเสริมเหล่านั้นใหม่ โดยเรียก Eclipse แบบนี้ว่า EasyEclipse โดยชุดโปรแกรมของEasyEclipse มีหลายรูปแบบเช่น

* EasyEclipse Expert Java สำหรับผู้ที่เขียนโปรแกรม Java แบบมืออาชีพ
* EasyEclipse Desktop Java สำหรับผู้ที่พัฒนาโปรแกรม Java แบบเน้นไปที่โปรแกรมบน Desktop
* EasyEclipse Server Java สำหรับผู้ที่พัฒนาโปรแกรม Java แบบเน้นไปที่โปรแกรมด้าน Server
* EasyEclipse Mobile Java สำหรับผู้ที่พัฒนาโปรแกรม Java แบบเน้นไปที่โปรแกรมบนมือถือ.Mobile device
* EasyEclipse for LAMP สำหรับผู้พัฒนาโปรแกรมบน LAMP\*
* EasyEclipse for PHP สำหรับผู้พัฒนาเว็บด้วย PHP
* EasyEclipse for Ruby and Rails สำหรับผู้พัฒนาเว็บด้วย Ruby on rail.
* EasyEclipse for C and C++ สำหรับผู้พัฒนาโปรแกรมด้วย C/C++

ในที่นี้ผู้พัฒนาได้เลือก EasyEclipse for LAMP มาใช้เป็นเครื่องมือสำหรับการพัฒนาเพราะว่า EasyEclipse for LAMP มีเครื่องมือที่ช่วยในการพัฒนาหลายอย่างรวมมาให้อยู่แล้ว ซึ่งเครื่องมือที่ผู้พัฒนาใช้หลักๆ ได้แก่ Pydev และ SVN ซึ่ง EasyEclipse for LAMP ยังมีความสามารถในการพัฒนา Ruby on rail, PHP และ Java อยู่ด้วย ทำให้ไม่จำเป็นต้องเปลี่ยนเครื่องมือบ่อยในการที่จะพัฒนาโปรแกรมต่างๆ ในแต่ล่ะครั้ง

## GraphML

GraphML เป็น xml ชนิดหนึ่งซึ่งออกแบบมาเพื่อการเก็บข้อมูลชนิดกราฟ ซึ่งออกแบบโดย Graphdrawing.org โดยการออกแบบนั้นครอบคลุมทุกเนื้อหา และง่ายที่จะนำมาใช้กับข้อมูลประเภทกราฟ ซึ่งมันประกอบไปด้วยส่วนประกอบหลักของตัวภาษาที่ใช้ในการอธิบายข้อมูลประเภทกราฟ และยืดหยุ่นพอสำหรับที่จะให้มีส่วนเสริมเพิ่มเข้าไปสำหรับข้อมูลเฉพาะของแต่โปรแกรมที่นำไปใช้ ซึ่งส่วนประกอบหลักของตัวภาษาจะประกอบไปด้วย

* กราฟแบบมีทิศทาง กราฟไม่มีทิศทาง กราฟแบบผสม
* Hypergraphs
* Hierarchical graphs
* Graphical representations
* References to external data
* Application-specific attribute data
* Light-weight parsers

ซึ่ง GraphML ได้ถูกนำไปใช้ในหลายๆ โปรแกรมที่ใช้ข้อมูลประเภทกราฟ และมีอีกหลายๆโปรแกรมที่ให้การสนับสนุนการใช้ GraphML เช่น JUNG, yFile, AT&T Research, IBM Research ซึ่งผู้พัฒนาได้นำเอาไฟล์ประเภทนี้มาใช้ในการเก็บข้อมูลประเภทกราฟ แต่ว่าได้นำความสามารถของ GraphML มาแค่บางส่วนเท่านั้น

## Graph

ในคณิตศาสตร์และวิทยาการคอมพิวเตอร์ กราฟ คือวัตถุพื้นฐานของการศึกษาในทฤษฎีกราฟ กล่าวอย่างไม่เป็นทางการได้ว่า, กราฟคือเซตของวัตถุที่เรียกว่า จุดยอด (vertex) ซึ่งเชื่อมต่อกันด้วย เส้นเชื่อม (edge) ในบางการประยุกต์ใช้งาน เส้นเชื่อมอาจแสดงอย่างมีทิศทางได้

**นิยาม**

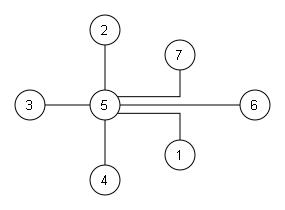
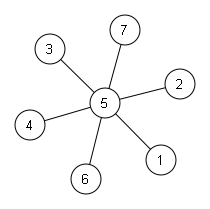
ในคณิตศาสตร์และวิทยาการคอมพิวเตอร์ กราฟ คือวัตถุพื้นฐานของการศึกษาในทฤษฎีกราฟ นิยามได้ว่ากราฟคือ เซทของความสัมพันธ์ระหว่างจุดตัด (Vertex, Node) ที่เชื่อมต่อกันด้วยเส้นเชื่อม (Edge) โดยที่การประยุกต์ใช้งานบางครั้ง เส้นเชื่อมอาจมีทิศทางได้

graph

**รูปที่ 2.1** อธิบายความหมายกราฟ

ถ้ามีข้อมูลเป็นเซทของความสัมพันธ์ระหว่างเส้นเชื่อม และจุดตัด ชุดหนึ่งๆก็สามารถที่จะนำมาเขียนเป็นกราฟได้หลายรูปแบบในการแสดงผลแม้ว่า จะเป็นข้อมูลชุดเดียวกันก็ตาม

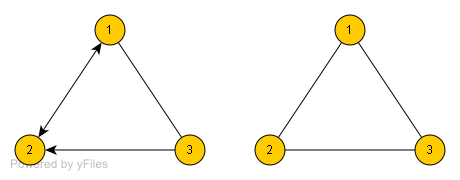
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |



Organic graph layout Orthogonal graph layout adjacency matrix

**รูปที่ 2.2** ความแตกต่างของการแสดงผลกราฟ

กราฟสามารถแบ่งชนิดได้ตามลักษณะของ เส้นเชื่อม อีกด้วยคือ กราฟแบบมีทิศทางและไม่มีทิศทาง

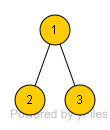
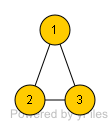


กราฟแบบมิทิศทาง กราฟแบบไม่มีทิศทาง

**รูปที่ 2.3** ความแตกต่างของกราฟแบบมีทิศทางและไม่มีทิศทาง

กราฟที่ เส้นเชื่อม มีทิศทางแม้เพียงเส้นเดียวก็ถือว่ากราฟนั้นเป็น กราฟที่มีทิศทางแล้ว ในที่นี้คือรูปตัวอย่างทางด้านซ้าย ส่วนทางด้านขวาคือกราฟไม่มีทิศทาง ซึ่งเรื่องของทิศทางนี้จะนำไปใช้พิจารณาในการจัดรูปแบบกราฟบางประเภท

กราฟยังแบ่งประเภทได้ด้วยการเชื่อมโยงของ เส้นเชื่อมได้อีกแบบคือ แบบต้นไม้(Tree) กับ แบบเครือข่าย (Network) โดย ต้นไม้คือกราฟที่ไม่เกิดการเชื่อมโยงเป็นวงขึ้นภายในตัวมันเองเลยและ แบบเครือข่ายคือกราฟที่มีการเชื่อมโยงกันเป็นวง ขึ้นภายในกราฟนั้นตั้งแต่ 1 วงขึ้นไป

ต้นไม้ เครือข่าย

**รูปที่ 2.4** ความแตกต่างของกราฟแบบต้นไม้และแบบเครือข่าย

**บทที่ 3**

วิธีการดำเนินโครงงาน

วิธีการดำเนินโครงงานระบบการจัดการผู้ใช้ มีวิธีในการดำเนินโครงงานดังต่อไปนี้ คือ

3.1 การศึกษาปัญหาของการแสดงผลข้อมูลประเภทกราฟแบบเดิม

3.2 การศึกษาความเป็นไปได้ และขอบเขตของการทำงาน

3.3 การวิเคราะห์ และออกแบบโปรแกรมแล้วโครงสร้างข้อมูล

**3.1 การศึกษาปัญหาของการแสดงผลข้อมูลประเภทกราฟแบบเดิม**

เนื่องจากข้อมูลในปัจจุบันมีข้อมูลเชิงความสัมพันธ์อยู่มากซึ่งข้อมูลเหล่านั้นสามารถแทนให้อยู่ในรูปของกราฟได้ แต่ว่าแม้ว่าข้อมูลจะอยู่ในรูปแบบของกราฟแต่การที่ จะคิดดำเนินการกับข้อมูลดังกล่าวนั้นปัจจุบันนั้นใช้การจินตนาการรูปกราฟต่างๆ นั้นเป็นส่วนมาก ซึ่งอาจจะทำให้ไม่เห็นภาพของข้อมูลที่ชัดเจน ซึ่งการจินตนาการรูปของกราฟนั้นอาจจะทำให้เกิดความผิดพลาดในการคิดได้ หรือว่าถ้าข้อมูลมีขนาดใหญ่การจินตนาการนั้นอาจจะไม่เพียงพอที่จะใช้ในการวิเคราะห์กราฟที่เกิดขึ้น หรือในการที่เราใช้ อัลกอริทึมที่จะใช้จัดการกับกราฟนั้นอาจจะทำให้ได้ผลที่เข้าใจได้ยาก และการจะอธิบายผลต่อผู้อื่นก็เป็นเรื่องยากด้วย

**3.2 การศึกษาความเป็นไปได้**

จากการศึกษาการทำงานกับข้อมูลชนิดกราฟแบบเดิม จะพบว่าปัญหาเกี่ยวกับการนำเสนอข้อมูลเชิงความสัมพันธ์(หรือกราฟ) ในรูปแบบเดิมนั้น ยังไม่มีประสิทธิภาพเพียงพอสำหรับการนำไป แสดงผลให้ผู้ใช้ทั่วไปสามารถเข้าใจได้ ทำให้การนำไปคิดวิเคราะห์และพัฒนาต่อเป็นไปได้ยาก เพราะว่าผู้ใช้ไม่เข้าใจข้อมูล การปรับการแสดงผลข้อมูลชนิดกราฟให้อยู่ในรูปแบบกราฟิก อาจจะเป็นแนวทางหนึ่งในการช่วยให้ผู้ที่ดูข้อมูลเชิงความสัมพันธ์ (หรือกราฟ) สามารถเข้าใจข้อมูลเหล่านั้นได้มากขึ้น

ผู้จัดทำโครงงานจึงมีแนวคิดที่จะพัฒนาโปรแกรมแสดงผลข้อมูลชนิดกราฟให้อยู่ในรูปกราฟิก ขึ้นเพื่อคอยให้บริการในส่วนของการแสดงผลข้อมูลเชิงความสัมพันธ์ต่างๆ ออกมาในรูปแบบกราฟิก โดยโปรแกรมจะมีความสามารถหลักๆดังนี้

1. สามารถอ่านไฟล์ข้อมูลเชิงความสัมพันธ์ขึ้นมาแสดงผลได้
2. สามารถที่จะสร้างหรือแก้ไขกราฟได้เอง เพื่อใช้ในการจัดการกราฟต่างๆ ได้ในระดับหนึ่ง เช่น การเพิ่มลดจุดตัด เคลื่อนย้ายจุดตัด การบันทึกกราฟหลังการเปลี่ยนแปลง
3. สามารถที่จะเพิ่มความสามารถในการจัดการกราฟด้วยอัลกอริทึมโดยจะแสดงผลออกมาโดยการเน้นสีที่ จุดตัดและเส้นที่เป็นผลลัพธ์
4. สามารถบันทึกกราฟที่แก้ไขหรือสร้างขึ้นมาได้โดยที่จะบันทึกแค่ส่วนของโครงสร้างเท่านั้น ไม่เก็บส่วนของตำแหน่งหรือสีของกราฟ

**3.3 การออกแบบโปรแกรม**

จากการศึกษาความเป็นไปได้ของการดำเนินการ แล้วจึงนำเข้าสู่ขั้นตอนของการออกแบบโปรแกรม โดยที่ผู้พัฒนาได้ทำการออกแบบโปรแกรมในรูปแบบของ Object Oriented โดยการมองส่วนประกอบของโปรแกรมเป็น Class และ Object เพื่อนำไปเขียนโปรแกรมต่อไปโดยส่วนประกอบหลักๆที่ต้องออกแบบมีดังนี้

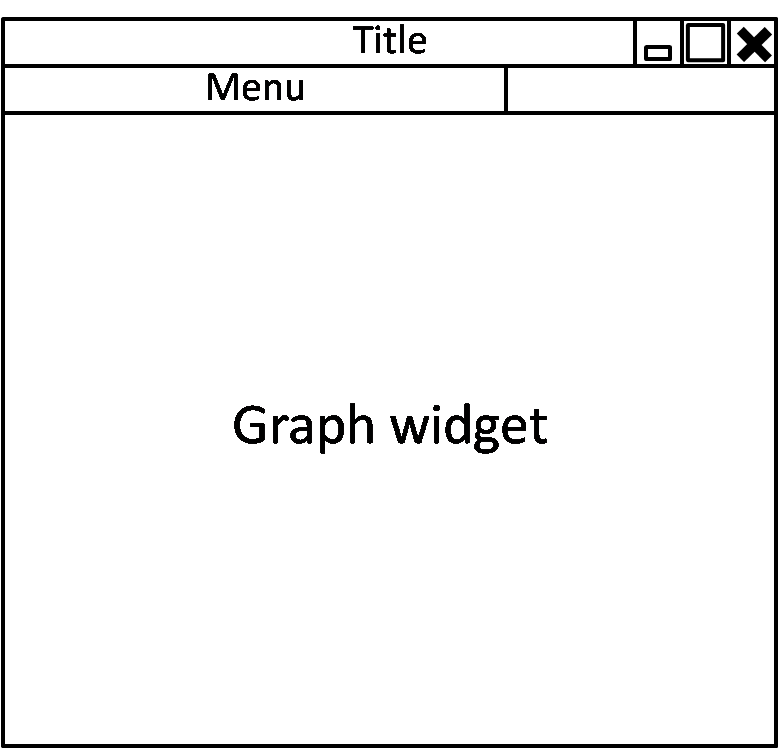
3.1.1 ออกแบบส่วนของการติดต่อกับผู้ใช้ (User Interface)

3.1.2 คลาสไดอะแกรม (Class Diagram)

**3.3.1 ออกแบบส่วนของการติดต่อผู้ใช้ (User Interface)**

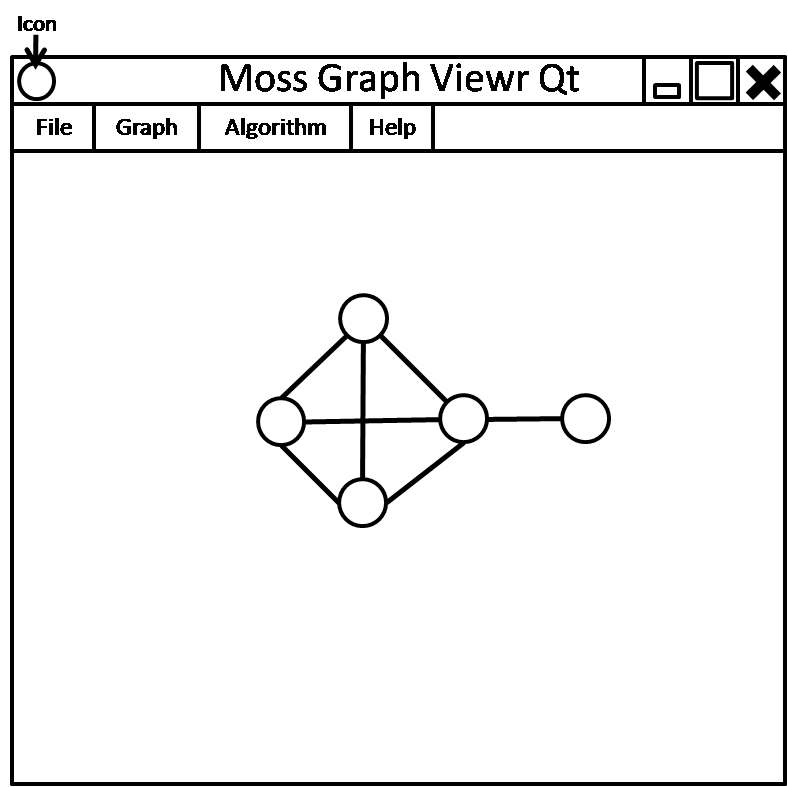
ออกแบบส่วนติดต่อผู้ใช้โปรแกรม เพื่อกำหนดส่งแนวทางในการพัฒนาโปรแกรม และสิ่งที่จำเป็นต้องมีในโปรแกรม

1. โครงสร้างหน้าตาหลักๆของโปรแกรมจะมีลักษณะดังรูป



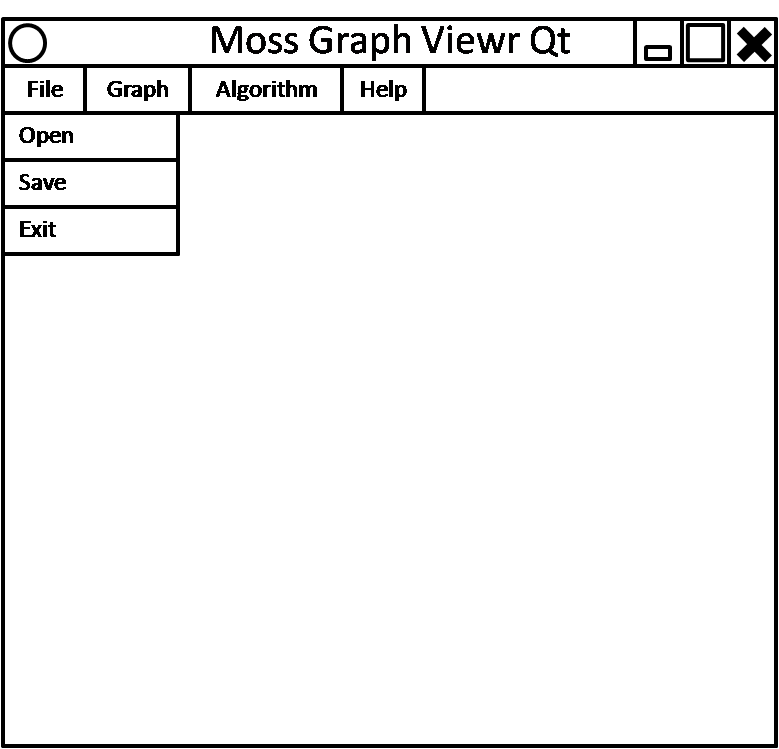
**รูปที่ 3-1** รูปโครงสร้างหลักของส่วนติดต่อผู้ใช้ของโปรแกรม

2. รูปร่างหน้าตาของโปรแกรมเมื่อใช้งานซึ่งประกอบไปด้วยการนำข้อมูลกราฟที่เปิดจากไฟล์ขึ้นมาแทนที่ Graph widget เมนูต่างๆ ที่จำเป็นต้องมี ชื่อโปรแกรมและไอคอนตกแต่ง



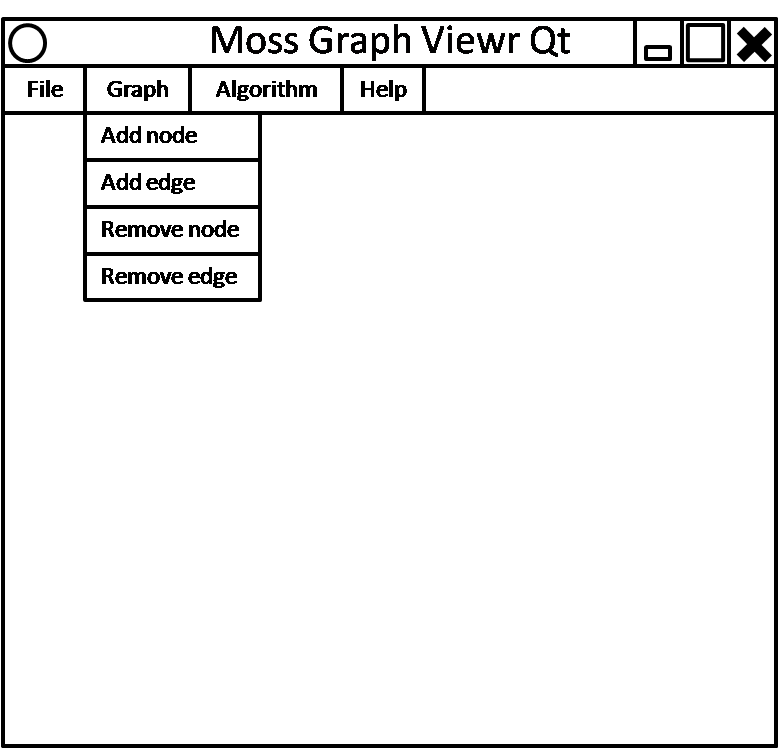
**รูปที่ 3-2** รูปส่วนติดต่อผู้ใช้ของโปรแกรมเมื่อใช้งาน

3. ส่วนติดต่อผู้ใช้เกี่ยวกับไฟล์จะมีความสามารถพื้นฐานในการเปิดไฟล์ การบันทึกไฟล์และออกจากโปรแกรม ส่วนหน้าต่างในการหาไฟล์(File manager, File Browser) จะเรียกใช้ของระบบโดยตรง (ขึ้นอยู่กับระบบปฏิบัติการ)



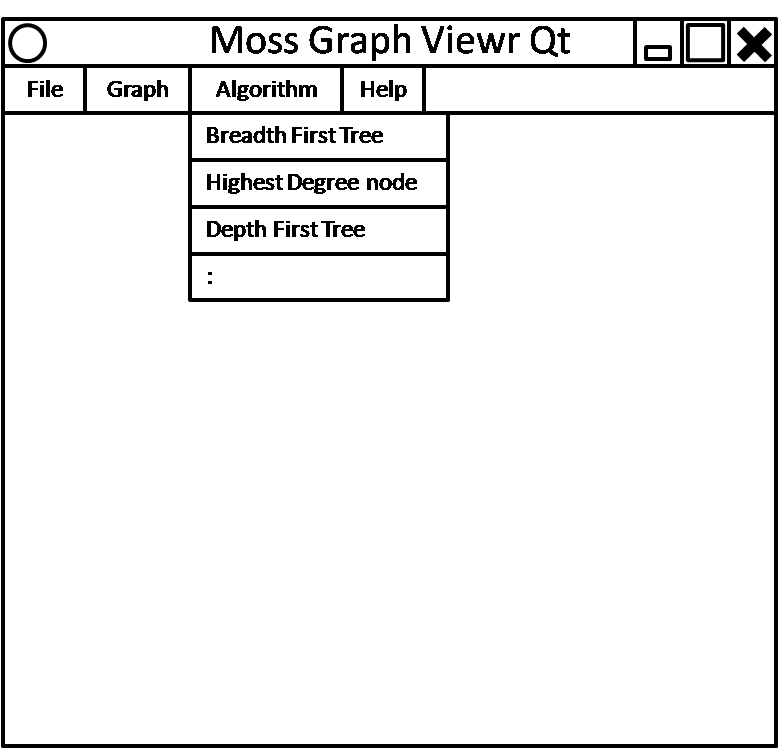
**รูปที่ 3-3** รูปส่วนติดต่อผู้ใช้ เมนูไฟล์

4. ส่วนติดต่อผู้ใช้ เมนูกราฟใช้เพื่อเก็บการจัดการหลักๆเกี่ยวกับกราฟไว้เช่น การเพิ่มลดจุดตัด(Node) การเพิ่มลดเส้นเชื่อม ( Edge)



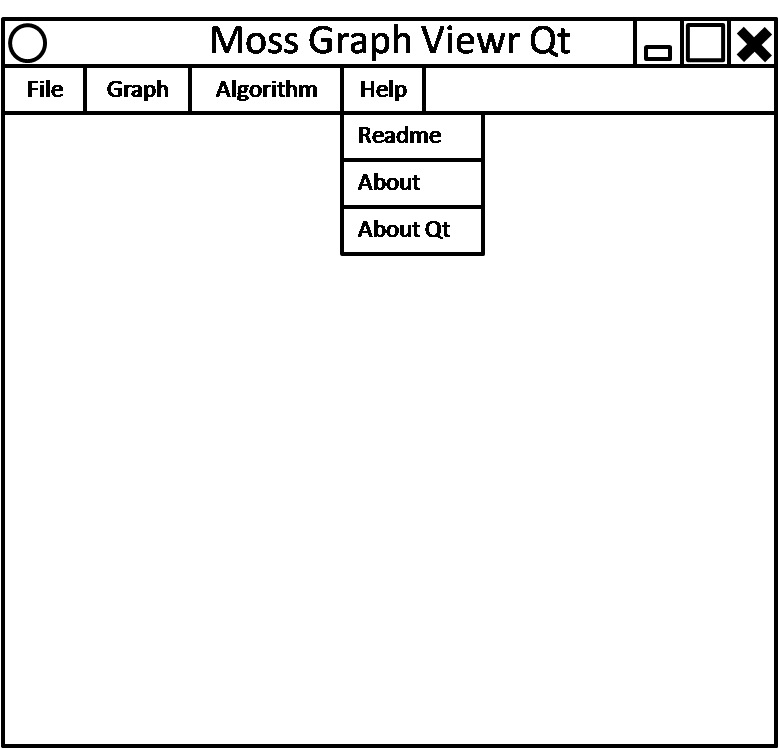
**รูปที่ 3-4** รูปส่วนติดต่อผู้ใช้ เมนูกราฟ

5. ส่วนติดต่อผู้ใช้ เมนูอัลกอริทึ่ม ตัวเลือกสำหรับการใส่การจัดการอัลกอริทึ่มซึ่งสามารถใส่เพิ่มขึ้นเข้าไปได้เรื่อยๆ โดยการแก้ไขปรับปรุงโปรแกรมส่วนนี้

****

**รูปที่ 3-5** รูปส่วนติดต่อผู้ใช้ เมนูอัลกอริทึม

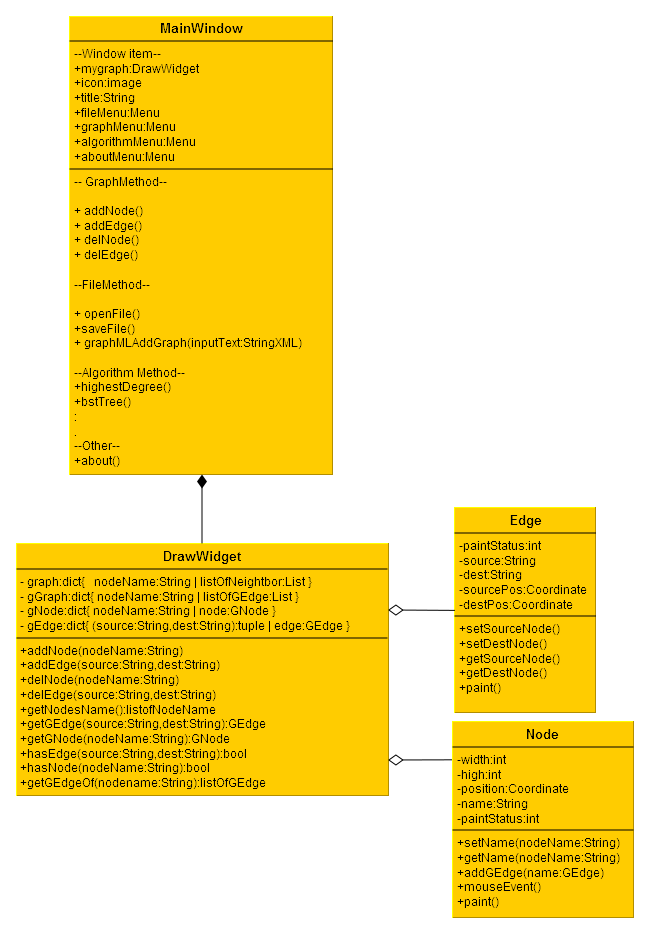
6. ส่วนติดต่อผู้ใช้ เมนูช่วยเหลือและเกี่ยวกับ ใช้สำหรับใส่ส่วนช่วยเหลือผู้ใช้ และแสดงข้อมูลเกี่ยวกับโปรแกรมและการพัฒนา



**รูปที่ 3-6** รูปส่วนติดต่อผู้ใช้ เมนูอัลกอริทึม

**3.3.2 คลาสไดอะแกรม (Class Diagram)**

คลาสไดอะแกรมต้นแบบของโปรแกรมแสดงผลข้อมูลประเภทกราฟออกมาในรูปแบบกราฟิก ได้ออกแบบส่วนประกอบสำคัญ ไว้เพื่อเป็นต้นแบบในการเขียนโปรแกรมจริง แสดงให้เห็นดังรูป



**รูปที่ 3-7** คลาสไดอะแกรมของโปรแกรมแสดงผลข้อมูลประเภทกราฟ

**บทที่ 4**

**ผลการดำเนินโครงงาน**

จากการออกแบบโปรแกรมแสดงผลข้อมูลประเภทกราฟ และได้นำมาพัฒนา จนได้เป็นโปรแกรมแสดงผลข้อมูลประเภทกราฟซึ่งมีส่วนประกอบหลักๆ ที่สามารถนำมาแสดงผลให้เห็นได้ดังนี้

4.1 ส่วนแสดงผลกราฟ

4.2 ส่วนจัดการกับกราฟ

4.3 ส่วนดำเนินการกับไฟล์

4.4 ส่วนอัลกอริทึม

**4.1 ส่วนแสดงผลกราฟ**

ส่วนประกอบหลักสำคัญที่สุดของโปรแกรมคือส่วนของการแสดงผลกราฟ จัดอยู่ที่ส่วนแสดงผลหลักของโปรแกรมเพื่อรองรับการจัดการกับกราฟต่างๆ ซึ่งมีความสามารถและข้อจำกัดดังนี้

1. สามารถใช้เมาส์ในการจัดวางตำแหน่งของจุดตัดของกราฟได้

2. สามารถกำหนดจุดตัดที่เน้นหรือเส้นเชื่อมที่เน้นได้

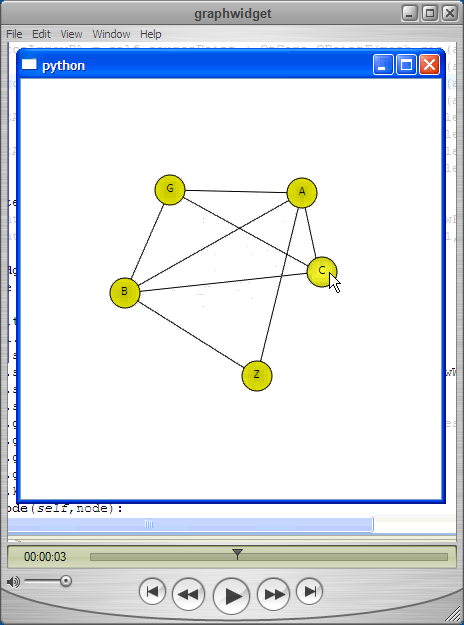
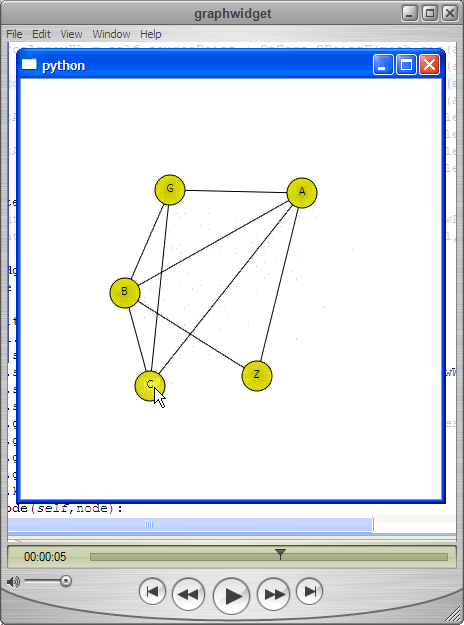
3. สามารถแสดงแบบย่อ/ขยายได้ (Zoom)

4. รูปร่างของจุดตัดและเส้นเชื่อมไม่สามารถเปลี่ยนแปลงได้

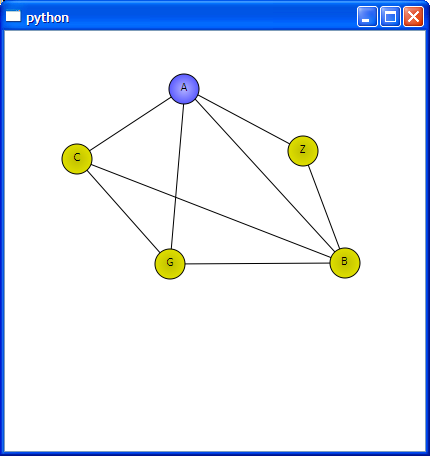
5. จะถูกแทนที่ถ้ามีการโหลดกราฟใหม่เข้ามาแทนที่

6. ชื่อที่กำหนดลงในแต่ละจุดตัดกำหนด จำกัดความยาวเพียง 3 ตัวอักษร

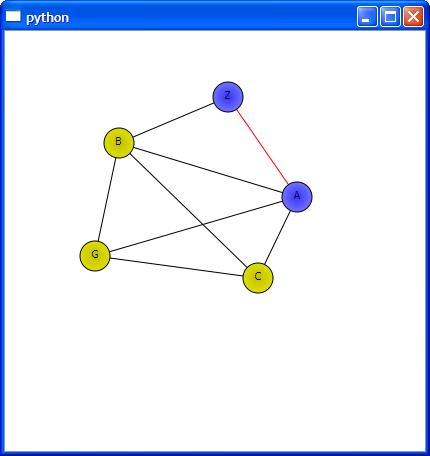
7. สีของจุดตัดและเส้นเชื่อมถูกกำหนดไว้แล้ว

**รูปที่ 4-1** รูปการเลื่อนจุดตัดได้ของโปรแกรม (ในรูปคือการเลื่อนจุดตัด C)



**รูปที่ 4-2** รูปการเน้นจุดตัด (A)



**รูปที่ 4-3** รูปการเน้นเส้นเชื่อม (A-Z)

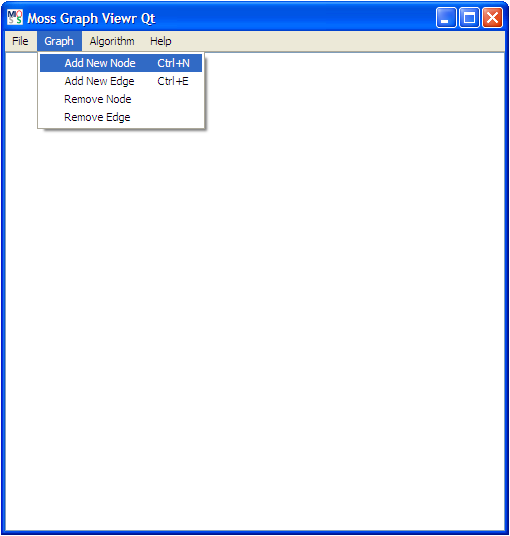
**4.2 จัดการกับกราฟ**

ส่วนที่ใช้ในการจัดการกับข้อมูลของกราฟเช่นการเพิ่มลดขนาดของกราฟ ทำให้มีความยืดหยุ่นในการปรับแก้หรือว่าทดสอบกราฟต่างๆ ผู้จัดทำได้จึงได้เพิ่มความในการจัดการกับกราฟขึ้นมา 4 อย่าง คือ

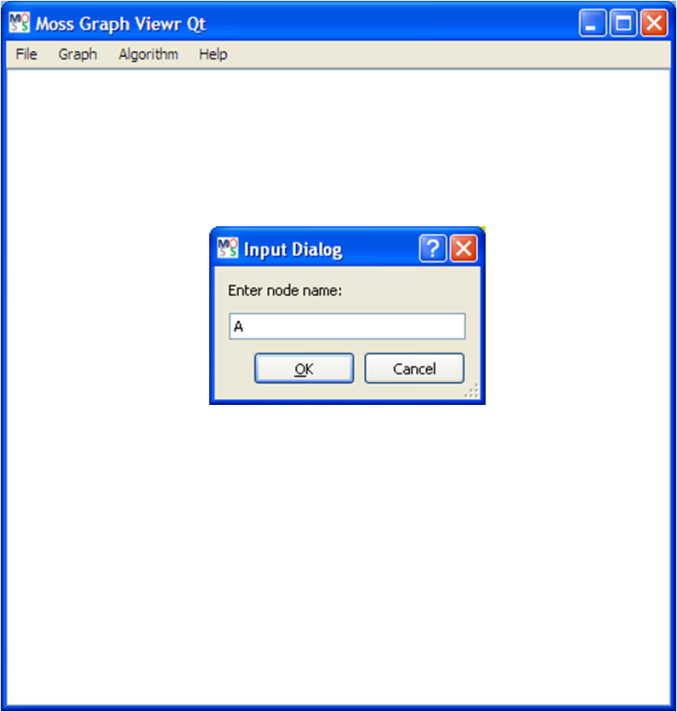
1. การเพิ่มจุดตัด
2. การเพิ่มเส้นเชื่อม
3. การลบจุดตัด
4. การลบเส้นเชื่อม

**4.2.1 การเพิ่มจุดตัด**

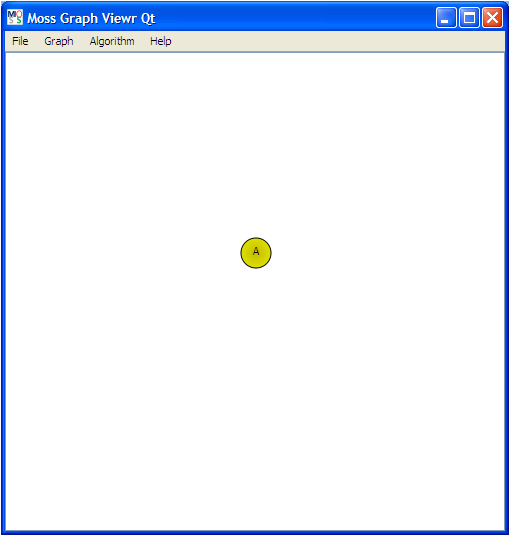
การเพิ่มจุดตัดหรือการเพิ่ม Node กำหนดให้ต้องมีการกำหนดชื่อเพื่อที่จะให้มีตัวกำหนดความแต่ต่างระหว่างแต่ล่ะจุดตัด ผู้พัฒนาได้สร้างวิธีในการเพิ่มจุดไว้สองแบบคือการใช้คีย์ลัดและเมนู



**รูปที่ 4-4.1** รูปตัวอย่างการเพิ่มจุดตัด (Node)



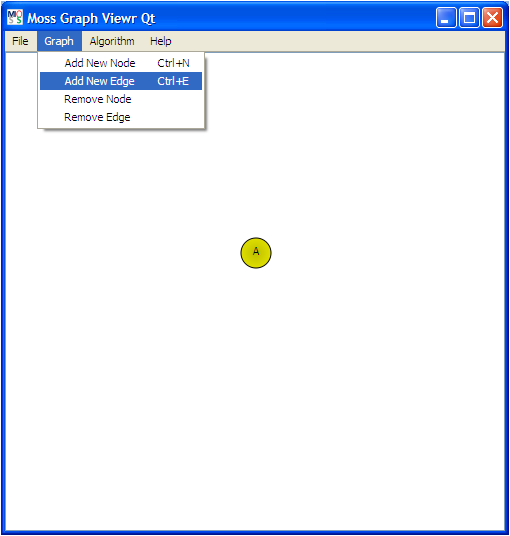
**รูปที่ 4-4.2** รูปตัวอย่างการเพิ่มจุดตัด (Node)



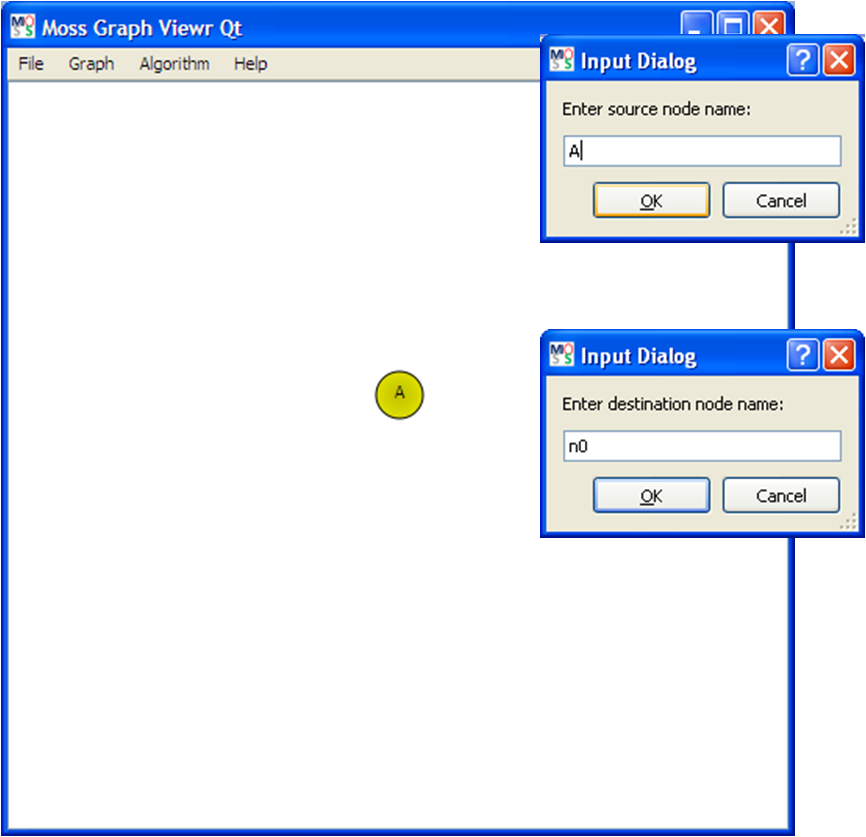
**รูปที่ 4-5** รูปตัวอย่างผลของการเพิ่มจุดตัด (Node)

**4.2.2 การเพิ่มเส้นเชื่อม**

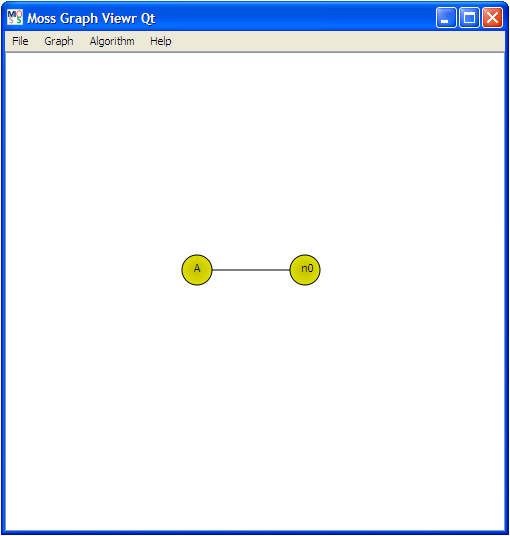
การเพิ่มเส้นเชื่อมหรือการเพิ่ม Edge กำหนดให้ต้องมีการกำหนดชื่อของสองจุดตัดที่จะใช้ในการเชื่อมต่อกัน ผู้พัฒนาได้สร้างวิธีในการเพิ่มจุดไว้สองแบบคือการใช้คีย์ลัดและเมนู เช่นเดียวกับการเพิ่มจุดตัด แต่ว่าถ้าเกิดมีการเพิ่มเส้นเชื่อมไปยังจุดตัดที่ยังไม่มีอยู่โปรแกรมจะสร้างจุดตัดนั้นให้โดยอัตโนมัติ



**รูปที่ 4-6** รูปตัวอย่างผลของการเลือกเมนูเพิ่มเส้นเชื่อม (Edge)



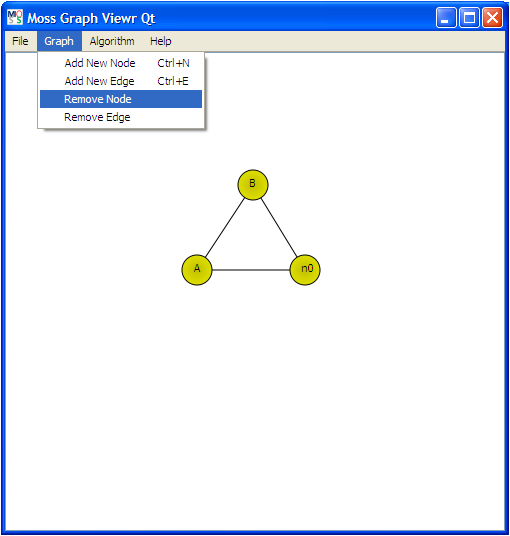
**รูปที่ 4-7** รูปตัวอย่างผลของการเพิ่มเส้นเชื่อม กรอกจุดเริ่มต้นและจุดปลาย



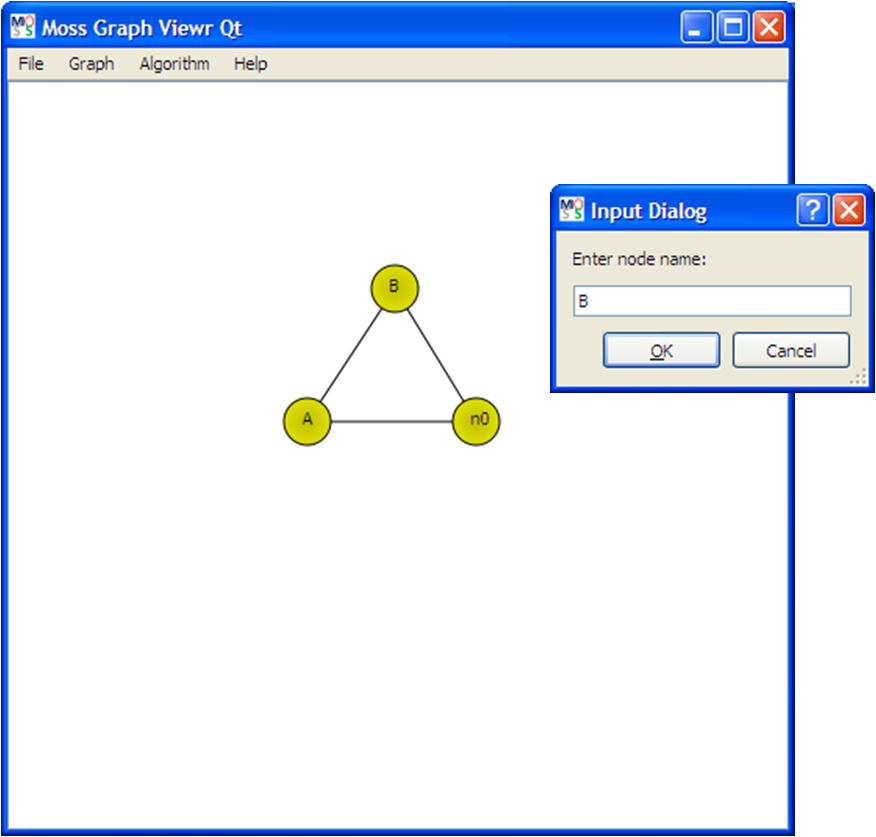
**รูปที่ 4-8** รูปตัวอย่างผลของการเพิ่มเส้นเชื่อม

**4.2.3 การลบจุดตัด**

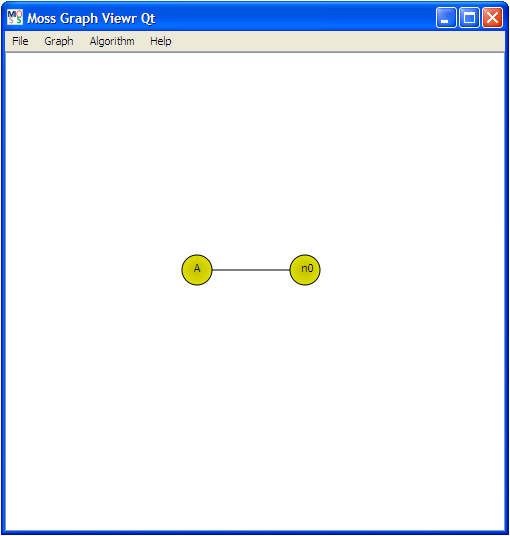
การลบจุดตัดจะต้องมีการกรอกชื่อของจุดตัดที่จะลบ เมื่อมีการลบจุดตัดออกจะต้องมีการพิจารณาว่าจุดตัดนั้นมีเส้นเชื่อมที่เชื่อมมาถึงมันหรือไม่ ถ้ามีเส้นเชื่อมที่เชื่อมมาถึงมัน ก็ต้องลบเส้นเชื่อมนั้นๆออกด้วย เพราะว่าเส้นเชื่อมแต่ล่ะเส้นเชื่อมจำเป็นต้องมีจุดปลายสองจุด



**รูปที่ 4-9** รูปตัวอย่างของการเลือกเมนูลบจุดตัด



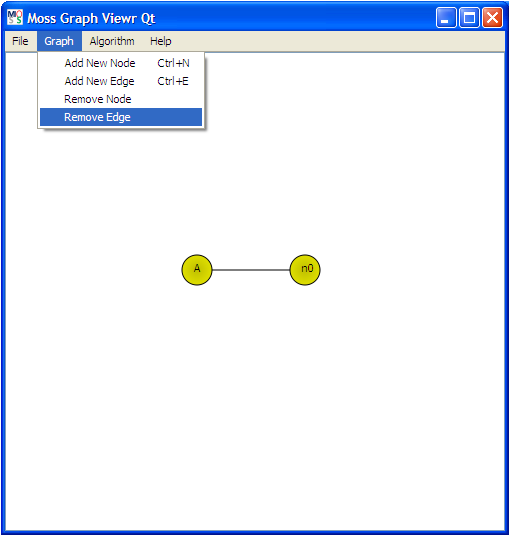
**รูปที่ 4-10** รูปตัวอย่างของการกรอกจุดตัดที่ต้องการลบ



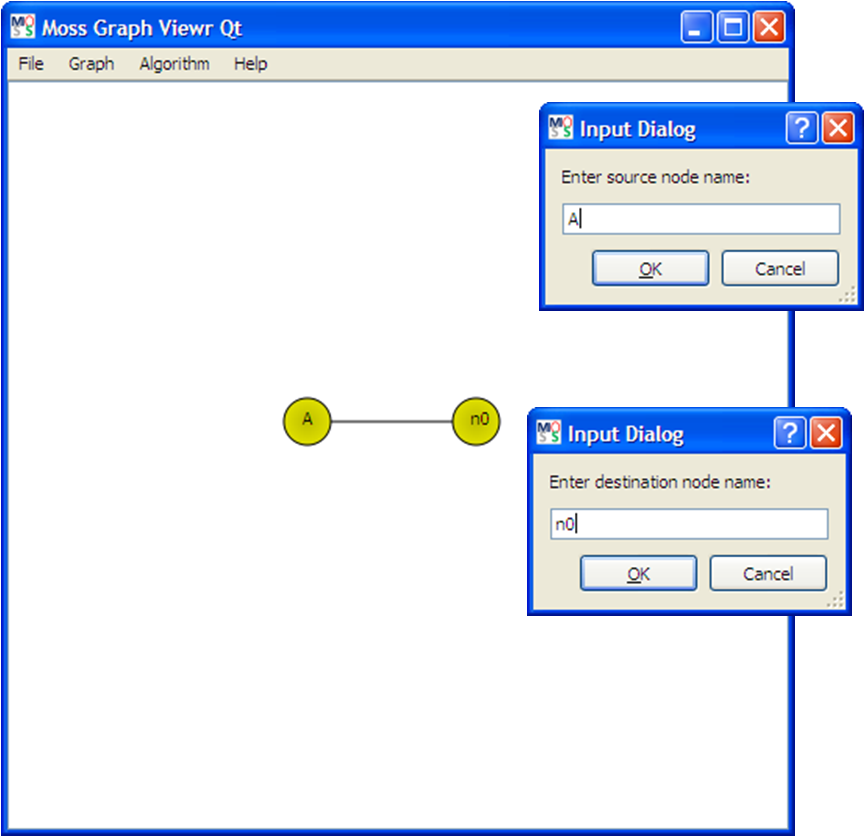
**รูปที่ 4-11** รูปตัวอย่างผลของการเลือกเมนูลบจุดตัด

**4.2.4 การลบจุดเส้นเชื่อม**

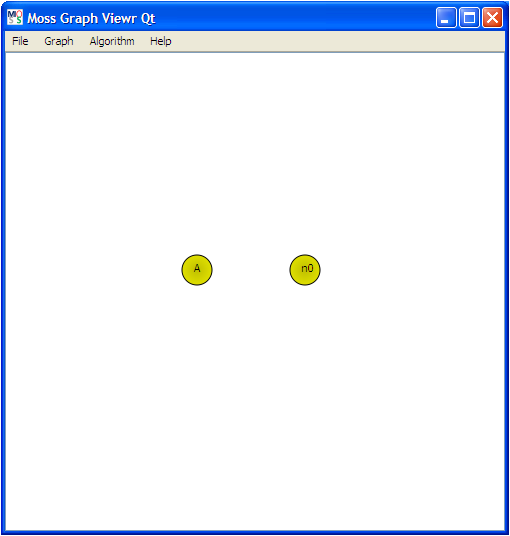
การลบเส้นเชื่อมจะต้องมีการกรอกชื่อของจุดปลายทั้งสองของเส้นเชื่อมที่จะลบ แต่ว่าการเส้นเชื่อมนั้นจะไม่มีผลใดๆ กับจุดตัดใดๆ ทั้งสิ้น



**รูปที่ 4-12** รูปตัวอย่างผลของการเลือกเมนูลบเส้นเชื่อม



**รูปที่ 4-13** รูปตัวอย่างผลของกรอกข้อมูลจุดปลายเพื่อลบเส้นเชื่อม



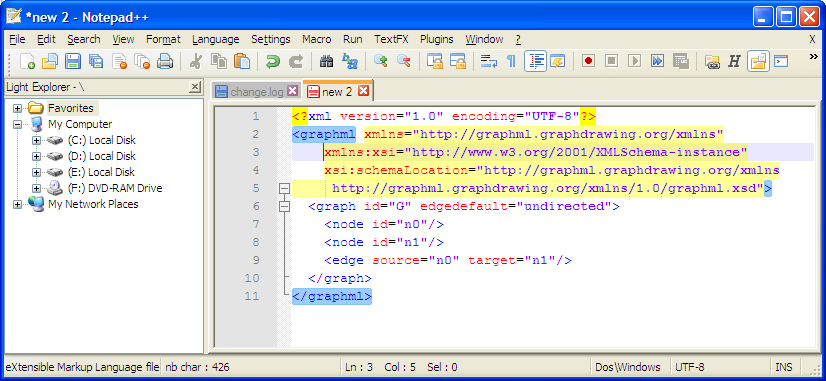
**รูปที่ 4-14** รูปตัวอย่างผลของการเลือกเมนูลบจุดตัด

**4.3 ส่วนดำเนินการกับไฟล์**

ส่วนที่ใช้ในการดำเนินการกับไฟล์เพื่อให้สามารถเก็บข้อมูลกราฟที่สร้างขึ้นมา หรือว่าเปิดอ่านข้อมูลกราฟที่ได้มาจากโปรแกรมอื่นที่ใช้ มาตรฐานเดียวกันในการเก็บข้อมูล ขึ้นมาแสดงผลได้

**4.3.1 การอ่านข้อมูลจากไฟล์**

ไฟล์ที่ใช้ในการเก็บข้อมูลของโปรแกรมนี้ใช้รูปแบบของ GraphML ซึ่งเป็นมาตรฐาน XML ตัวหนึ่งซึ่งใช้ในการเก็บข้อมูลประเภทกราฟโดยเฉพาะซึ่งมีโครงสร้างหลักๆ ดังนี้



**รูปที่ 4-15** รูปตัวอย่างไฟล์ GraphML

จะเห็นได้ว่าจะมีการประกาศ Namespace โดยแท็ก GraphML แต่ว่าโครงสร้างของกราฟจริงๆ นั้นเริ่มขึ้นที่แท็ก graph ซึ่งเป็นเสมือนเป็นจุดเริ่มต้นของข้อมูลกราฟซึ่งจะกำหนดด้วยว่ากราฟจะมีลักษณะเป็นแบบได้ เช่นมีทิศทาง (directed graph) หรือว่าเป็นกราฟที่ไม่มีทิศทาง (undirected graph) ซึ่งในโปรแกรมรุ่นปัจจุบันจะยังไม่พิจารณาเรื่องทิศทางของกราฟ

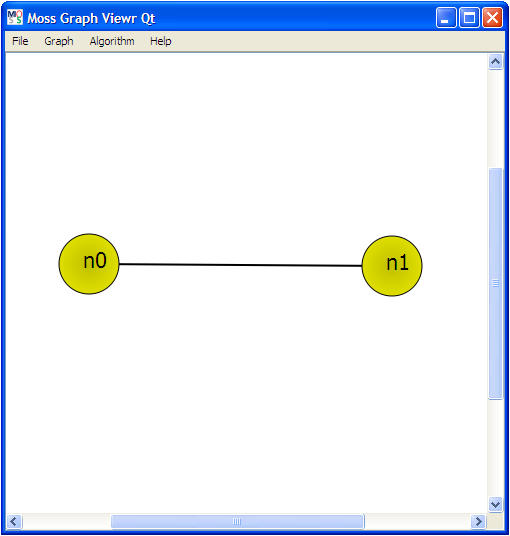
ส่วนที่กำหนดโครงสร้างของกราฟคือแท็ก node และ edge โดยที่แท็ก node จะต้องกำหนด id เสมอซึ่งสามารถที่จะกำหนดอย่างอื่นเพิ่มเติมภายในแท็กนั้นๆ ได้ตาม ความต้องการของโปรแกรมต่างๆ ที่อาจจะมีส่วนเสริมเพิ่มเติมในเรื่องนี้ ดูตัวอย่างได้ที่รูปที่ 4-16



**รูปที่ 4-16** ตัวอย่างไฟล์ GraphML ที่ใช้กับโปรแกรม yWork ที่เพิ่มส่วนเสริมเข้าไป

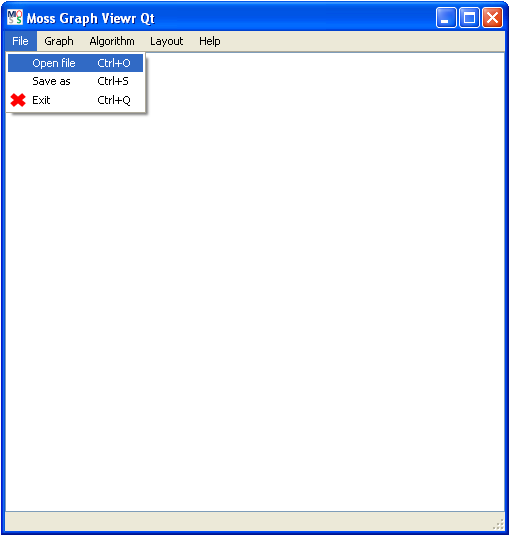
(ในกรอบคือส่วนเสริม)

แท็ก edge กำหนดการเชื่อมโยงด้วย แอตทริบิวต์ source และ target เสมอ แม้ว่าจะเป็นข้อมูลกราฟแบบ ไม่มีทิศทางก็ตาม เพื่อป้องกันความสับสนของผู้ใช้ที่ต้องนำไปใช้กับกราฟหลายแบบ และข้อมูลของแอตทริบิวต์ของ source และ target ก็คือ id ของ node ซึ่งโปรแกรมในส่วนของการดำเนินการกับไฟล์ของโปรแกรมนี้จะใช้ id ของแต่ล่ะ node มาเป็นชื่อของจุดตัด ตัวอย่างการแสดงผลข้อมูลกราฟข้างบนด้วยโปรแกรมจะแสดงออกมาดังรูปที่ 17

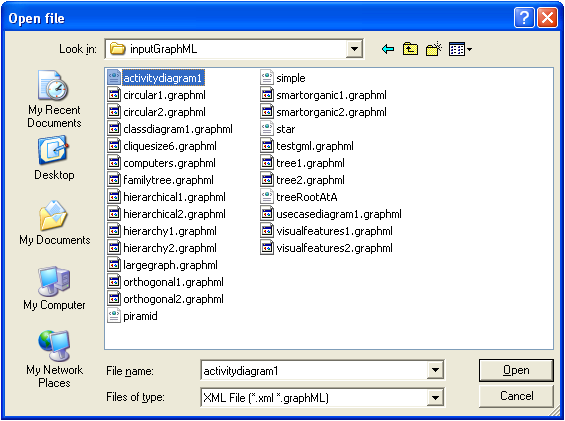


**รูปที่ 4-17** รูปตัวอย่างการแสดงผลข้อมูลกราฟจากไฟล์

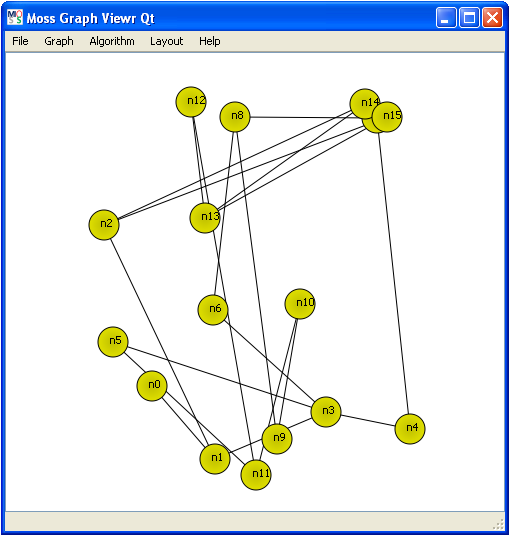
การเปิดหน้าต่างเลือกไฟล์เนื่องจาก PyQt มีความสามารถในการเรียกหน้าต่างเลือกไฟล์ของระบบปฏิบัติการ และสร้างตัวกรอกได้ทำให้ป้องกันปัญหาเรื่องการเปิดไฟล์ผิดรูปแบบได้ในระดับหนึ่ง ลำดับการเปิดเลือกไฟล์เพื่อนำมาแสดงผลดูได้ ในรูปที่ 4-18 ถึง 4-20 ตามลำดับ



**รูปที่ 4-18** รูปตัวอย่างการเลือกเมนูเปิดไฟล์



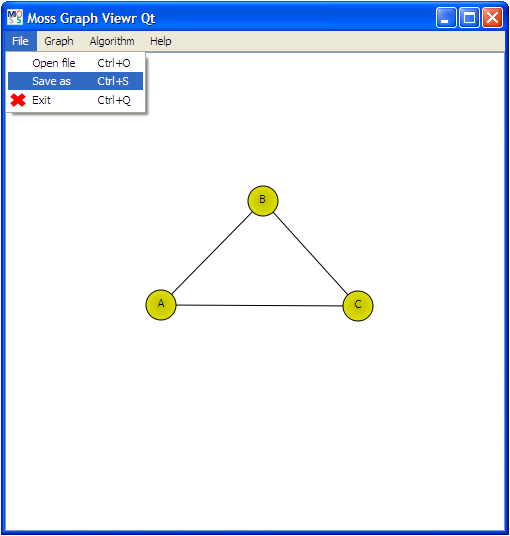
**รูปที่ 4-19** รูปตัวอย่างหน้าต่างเลือกเปิดไฟล์และตัวกรองไฟล์



**รูปที่ 4-20** รูปตัวอย่างไฟล์ที่ถูกเปิดขึ้นมาแสดงผล

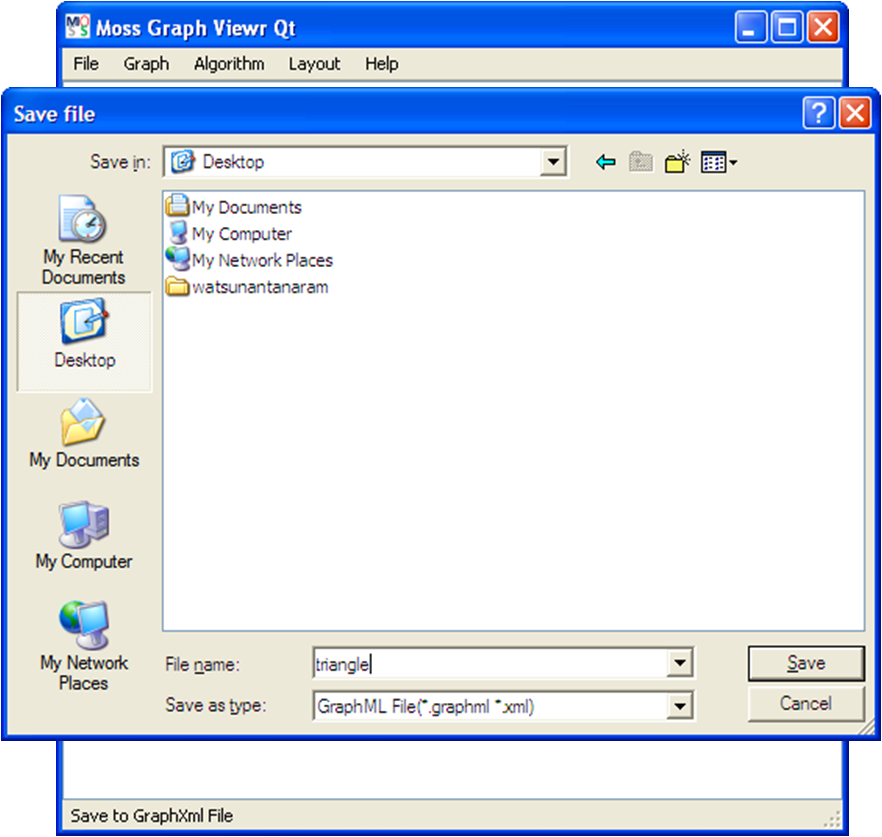
**4.3.2 การเก็บข้อมูล**

การที่จะเก็บข้อมูลของกราฟที่แก้ไขแล้วกลับไปเป็นไฟล์เพื่อที่จะสามารถนำไปใช้ต่อได้ในครั้งต่อไปจะต้องนำข้อมูลกราฟที่แสดงอยู่นั้นออกมาเป็นข้อมูลแล้วเขียนออกมาเป็นไฟล์ในรูปแบบของ GraphML โดยการเก็บไฟล์ของโปรแกรมนี้จะใช้ชื่อของจุดตัดแทนใน แอตทริบิวต์ id ของแท็ก node โปรแกรมจะไม่เก็บพิกัดของจุดหรือว่าสีที่เน้นไว้ แต่ว่าจะเก็บแต่โครงสร้างหลักๆ ของกราฟเท่านั้น รูปต่อไปนี้จะแสดงการบันทึกกราฟเป็นไฟล์ของโปรแกรม



**รูปที่ 4-20** รูปการเลือกเมนูบันทึกไฟล์

จากรูปมีการสร้างกราฟขนาด 3 จุดตัดขึ้นมา และต้องการที่จะบันทึกเป็นไฟล์เก็บไว้ ซึ่งในส่วนของการบันทึกไฟล์ผู้จัดทำก็ได้ทำตัวกรองไฟล์ไว้เช่นกัน นามสกุลของไฟล์โดยปริยายกำหนดให้เป็น GraphML ซึ่งผู้ใช้สามารถเลือกที่จะบันทึกเป็น xml ก็ได้แต่ว่าภายในไฟล์จะมีเนื้อหาเหมือนกัน



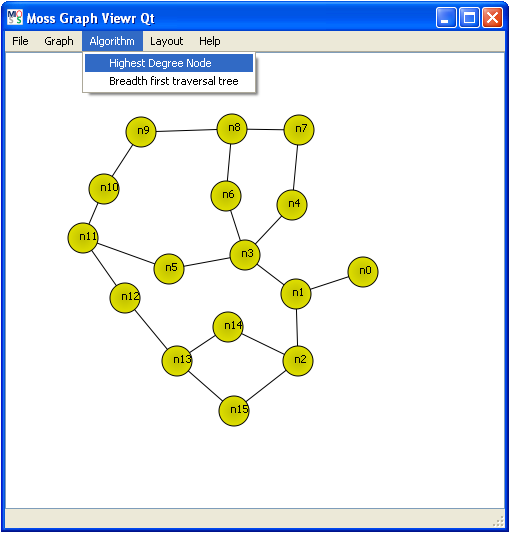
**รูปที่ 4-21** รูปหน้าต่างบันทึกไฟล์

เนื้อไฟล์ที่ได้จากการบันทึกกราฟข้างต้นจะได้ออกมาดังนี้

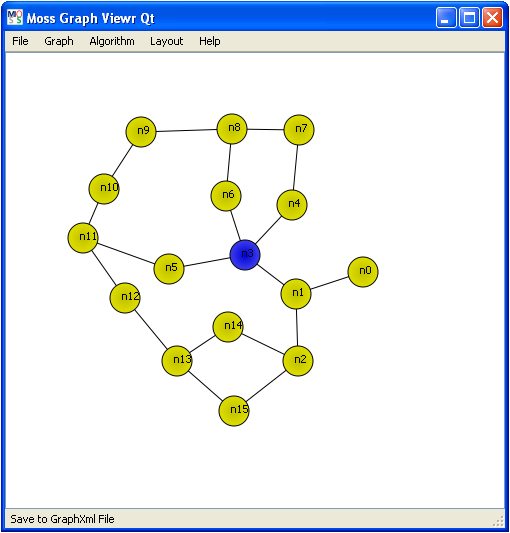
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">  <graph id="G" edgedefault="undirected">  <node id="A"/>  <node id="C"/>  <node id="B"/>  <edge source="A" target="B"/>  <edge source="C" target="B"/>  <edge source="A" target="C"/>  </graph>  </graphml> |

**4.4 ส่วนอัลกอริทึม**

ส่วนประกอบที่ทำให้เราสามารถนำกราฟอัลกอริทึมต่างๆ มากระทำกับกราฟแล้วทำให้มันแสดงผลออกมาในรูปแบบกราฟิกได้ โดยการที่เราจะเขียนโปรแกรมคำนวณเกี่ยวกับอัลกอริทึมแยกเป็นอีกส่วนหนึ่ง ของโปรแกรมเมื่อเรียกใช้ก็จะทำการส่งกราฟที่เป็นข้อมูลไปคำนวณและเมื่ออัลกอริทึม ทำงานเสร็จก็จะส่งผลการทำงานกลับมาซึ่งถ้าส่งกลับมาเป็นกราฟย่อยของกราฟที่ส่งไปข้อมูลไปก็จะทำการ เปลี่ยนสีกราฟที่เป็นผล หรือว่าจะแสดงผลเป็นแบบอื่นก็ขึ้นอยู่กับการทำงานในส่วนของแต่ล่ะอัลกอริทึมนั้นๆ แต่ว่าตอนนี้โปรแกรมสนับสนุนแค่การเน้นสีของกราฟเท่านั้น รูปที่ 4-22 และ รูปที่ 4-23 จะแสดงตัวอย่างการใช้อัลกอริทึมง่ายๆ อย่างการหาจุดตัดที่มีดีกรีสูงสุด

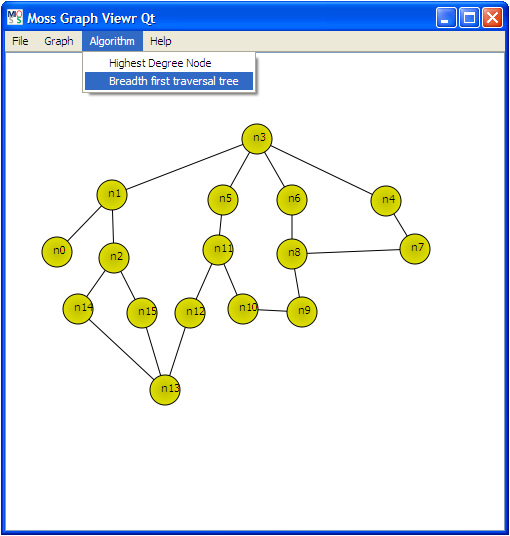


**รูปที่ 4-21** รูปกราฟต้นแบบก่อนการกระทำด้วยอัลกอริทึม

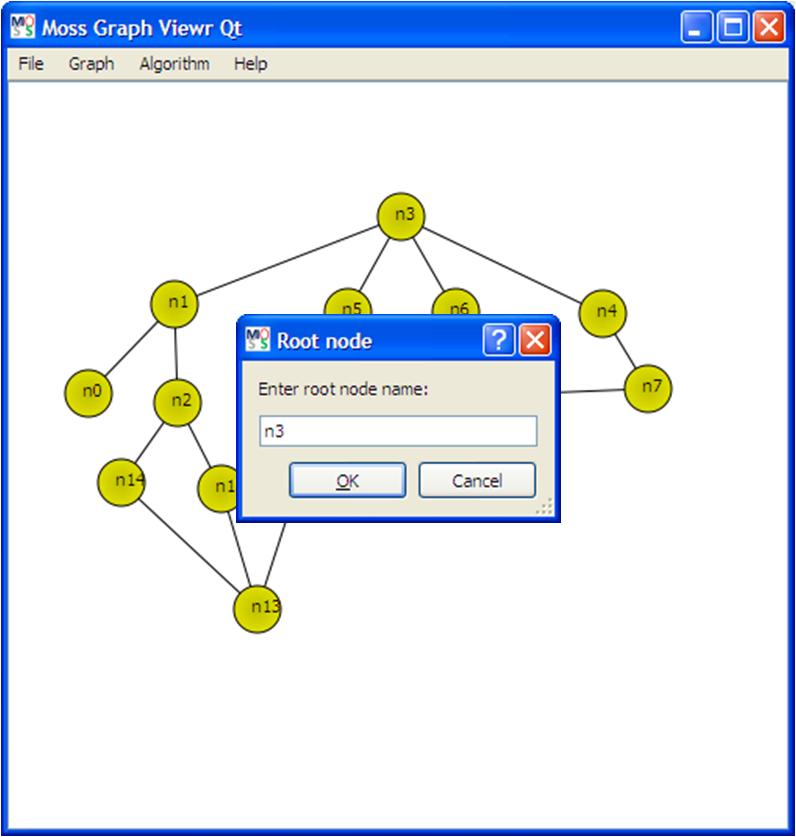


**รูปที่ 4-22** รูปผลลัพธ์ของการเลือกใช้อัลกอริทึมหาจุดตัดที่มีดีกรีสูงสุด

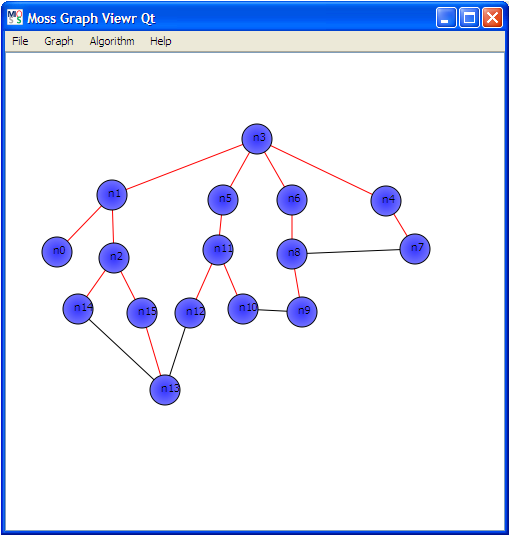
จากตัวอย่างข้างต้นคือ อัลกอริทึมหาจุดตัดที่มีดีกรีสูงสุดซึ่งไม่ต้องการข้อมูลอื่นนอกจากกราฟ แต่ว่าสำหรับในบางอัลกอริทึมแล้วต้องการข้อมูลเข้ามากกว่านั้นเช่น อัลกอริทึมในการหาต้นไม้ที่อยู่ในกราฟอย่าง Breadth first tree หรือว่า Breadth first search ที่ต้องการข้อมูลเข้าเป็น root ของต้นไม้และ node ที่ต้องการหา ดังนั้นจึงต้องมีการให้โปรแกรม รับข้อมูลเข้าที่เกี่ยวข้องก่อนที่จะทำงานได้ รูปที่ 4-23 และ รูปที่ 4-25 จะแสดงการผลของการใช้อัลกอริทึมที่ต้องการข้อมูลเข้าอื่นๆ โดยปัจจุบันโปรแกรมใช้วิธีการรับข้อมูลเข้าจากกล่องกรอกข้อความ



**รูปที่ 4-23** รูปกราฟต้นแบบก่อนการกระทำด้วยอัลกอริทึม (กราฟเดียวกับรูปที่ 21)



**รูปที่ 4-24** รูปตัวอย่างกรอกรับข้อมูลเพื่อนำไปใช้ในอัลกอริทึม



**รูปที่ 4-25** รูปตัวอย่างการเน้นกราฟที่ได้จากการกระทำของ อัลกอริทึม Breadth first trees

**บทที่ 5**

**สรุปและวิจารณ์ผลการดำเนินโครงงาน**

จากผลการดำเนินงานการพัฒนาโปรแกรมแสดงผลข้อมูลกราฟออกมาในรูปแบบกราฟิก สามารถสรุป และวิจารณ์ผลการดำเนินโครงการได้ดังนี้ คือ

5.1 สรุปผลการพัฒนาโปรแกรมและข้อจำกัดในแต่ล่ะส่วนของโปรแกรม

5.2 ข้อเสนอแนะในการพัฒนาโปรแกรมรุ่นต่อไป

**5.1 สรุปผลการพัฒนาโปรแกรมและข้อจำกัดในการทำงานของโปรแกรม**

สรุปผลการดำเนินโครงงานโปรแกรมแสดงผลข้อมูลกราฟออกมาในรูปแบบกราฟิก สามารถสรุปเป็นส่วนๆ ของโปรแกรมได้ดังนี้

1. ส่วนการแสดงผล
2. ส่วนการจัดการกับกราฟ
3. ส่วนการดำเนินการกับไฟล์
4. ส่วนอัลกอริทึม
5. ความสามารถการนำไปใช้ของโปรแกรม

**5.1.1 สรุปผลการทำงานส่วนการแสดงผลข้อมูลกราฟ**

การพัฒนาโปรแกรมในส่วนของการแสดงผลกราฟ สามารถสรุปผลการทำงาน ดังต่อไปนี้ คือ

1. สามารถแสดงผลกราฟออกในรูปแบบกราฟิกได้
2. สามารถเปลี่ยนตำแหน่งของจุดตัดต่างๆ ในกราฟได้
3. สามารถให้สีกับกราฟทั้งจุดตัดและเส้นเชื่อมได้
4. สามารถเพิ่มลดจุดตัดบนกราฟได้
5. สามารถใส่ชื่อของจุดตัดลงบนจุดตัดได้
6. สามารถซูมเข้าออกหน้าจอแสดงผลได้

ข้อจำกัดของโปรแกรมในส่วนของการแสดงผลข้อมูลกราฟได้แก่

1. ไม่สามารถเลือกจุดตัดหลายๆ จุดพร้อมกันได้
2. ไม่สามารถปรับแต่งสีทีให้กับกราฟโดยผู้ใช้ได้
3. ชื่อของจุดตัดที่ยาวเกิด 3 ตัวอักษรยังมีปัญหาในการแสดงผล
4. ไม่สามารถเปลี่ยนลักษณะของเส้นเชื่อมได้
5. ไม่สามารถเลื่อนส่วนหน้าจอแสดงผลได้(Pan)

**5.1.2 สรุปผลการทำงานส่วนการจัดการกับกราฟ**

การพัฒนาโปรแกรมในส่วนของการจัดการกับกราฟ สามารถสรุปผลการทำงาน ดังต่อไปนี้ คือ

1. เพิ่มลดจุดตัดได้
2. เพิ่มลดเส้นเชื่อมได้
3. การเพิ่มลดจุดตัดมีการจัดการกับเส้นเชื่อมให้

ข้อจำกัดของโปรแกรมในส่วนของการจัดการกับกราฟ ได้แก่

1. ไม่มีความสามารถในการเปลี่ยนชื่อจุดตัด

**5.1.3 สรุปผลการทำงานส่วนการดำเนินการกับไฟล์**

การพัฒนาโปรแกรมในส่วนของการดำเนินการกับไฟล์ สามารถสรุปผลการทำงาน ดังต่อไปนี้ คือ

1. สามารถอ่านไฟล์ประเภท GraphML ขึ้นมาแสดงผลเป็นกราฟิกได้
2. สามารถบันทึกไฟล์เป็นประเภท GraphML จากกราฟที่แสดงผลอยู่ปัจจุบันได้
3. สามารถอ่านไฟล์ประเภท GraphML ที่บันทึกมาด้วยโปรแกรมอื่นได้

ข้อจำกัดของโปรแกรมในส่วนของการดำเนินการกับไฟล์ ได้แก่

1. ไม่สามารถบันทึกพิกัดของจุดตัดลงได้
2. ไม่สามารถบันทึกรูปแบบของกราฟได้ เช่น สีของจุดตัดเส้นเชื่อมต่างๆ

**5.1.4 สรุปผลการทำงานส่วนอัลกอริทึม**

การพัฒนาโปรแกรมในส่วนอัลกอริทึม สามารถสรุปผลการทำงาน ดังต่อไปนี้ คือ

1. สามารถเพิ่มอัลกอริทึมในการจัดการกับกราฟได้ โดยที่ผลการดำเนินการที่ออกมาเป็นการเน้นสีที่กราฟ
2. สามารถเพิ่มอัลกอริทึมทีต้องการข้อมูลเข้าเพิ่มเติมได้(นอกจากโครงสร้างของกราฟ)

ข้อจำกัดของโปรแกรมในส่วนของการดำเนินการกับไฟล์ ได้แก่

1. การจะเพิ่มอัลกอริทึมหนึ่งๆ จำเป็นต้องแก้ไขโปรแกรมบางส่วน
2. ยังไม่สามารถเพิ่มอัลกอริทึมในการจัดวางกราฟได้(Graph Layout)

**5.1.4 สรุปผลความสามารถการนำไปใช้ของโปรแกรม**

ความสามารถในการนำไปใช้ของโปรแกรมนี้คือสามารถนำไปใช้กับระบบปฏิบัติการใดๆ ก็ได้ที่สามารถติดตั้ง Runtime ของทั้ง Python และ PyQt4 ได้ และมีการแสดงผลแบบกราฟิกได้ ซึ่งข้อจำกัดของโปรแกรมในส่วนนี้ก็คือ จำเป็นต้องมี Runtime ของทั้ง Python และ PyQt4 ติดตั้งอยู่ ถึงจะสามารถใช้งานได้

**5.2 ข้อเสนอแนะในการพัฒนาโปรแกรมรุ่นต่อไป**

ความสามารถที่ควรจะมีเพิ่มในการพัฒนาโปรแกรม เพื่อให้โปรแกรมมีประสิทธิภาพในการใช้งานที่ครอบคลุมมากยิ่งขึ้น สะดวกยิ่งขึ้น และควรจะพัฒนาในเบื้องต้นมีดังนี้

1. ความสามารถในการเลื่อนหน้าจอแสดงผล
2. ความสามารถในการเพิ่มอัลกอริทึมเพื่อใช้ในการจัดวางกราฟ (Graph Layout)
3. แก้ปัญหาเรื่องชื่อของจุดตัด
4. ความสามารถในการกำหนดสีของกราฟได้
5. ความสามารถในการบันทึกพิกัดของจุดตัดในกราฟ