ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนาโปรแกรมเสริมมายา
เพื่อสร้างภาพเคลื่อนไหวรูปปากการพูดภาษาไทย

A Development of a Maya Plug-in
for Lip Animation of Thai Language Speaking

H002363

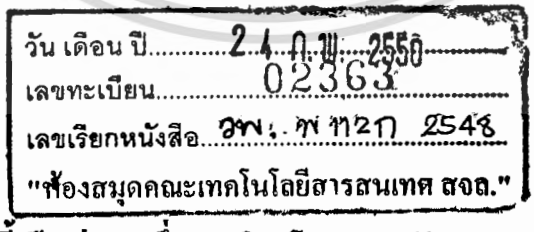
โดย

พงศ์รัฐ เมืองเจริญ

รหัสประจำตัว 47066144

อาจารย์ที่ปรึกษา

รศ.ดร.นพพร โชติกกำธร


วัน เดือน ปี 21 ก.พ. 2550
เลขทะเบียน 02363
เลขเรียกหนังสือ วพ. พ 1121 2548
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2548
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	**การพัฒนาโปรแกรมเสริมมายาเพื่อสร้างภาพเคลื่อนไหวรูปปากการพูดภาษาไทย**
นักศึกษา	**นายพงศ์รัฐ เมืองเจริญ**
อาจารย์ที่ปรึกษา	**รศ.ดร.นพพร โชติกกำธร**
ระดับการศึกษา	**วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ**
แขนงวิชา	**วิทยาการสารสนเทศ**
ปีการศึกษา	2548

บทคัดย่อ

ปัญหาสำคัญอย่างหนึ่งของโปรแกรมมายาคือ ในกระบวนการสร้างภาพตัวละครให้ขยับปากพูดนั้น เป็นกระบวนการที่ซับซ้อนและใช้เวลามาก โดยปกติจะเป็นหน้าที่ของผู้ใช้ในการสร้างตัวแบบรูปปากหลากหลายลักษณะขึ้นเองแล้วทำการปรับรูปปากเหล่านั้นให้มีรูปร่างตรงกับคำที่ต้องการพูด ดังนั้นโครงการนี้จึงนำเสนอแนวทางการแก้ปัญหาโดยพัฒนาโปรแกรมเสริมมายาขึ้น ทำให้ผู้ใช้เพียงโหลดตัวแบบและพิมพ์ข้อความที่ต้องการให้พูด จากนั้นโปรแกรมเสริมนี้จะแปลงข้อความนั้นให้เป็นหน่วยเสียงแล้วจับคู่เข้ากับรูปปากของตัวแบบ แล้วทำการผสมรูปปากหน่วยเสียงเป็นรูปปากเสียงพยางค์ ซึ่งวิธีการผสมรูปปากนี้สามารถทำโดยใช้ Blend Shape ภายในโปรแกรมมายา ในการสร้างภาพเคลื่อนไหวจะอาศัยวิธีการกำหนด Keyframe ของรูปปากของเสียงพยางค์ให้สัมพันธ์กับเวลาที่มีการออกเสียงพยางค์นั้น โปรแกรมเสริมนี้แบ่งเป็น 5 โมดูลสำคัญคือ โมดูลหลัก โมดูลส่วนติดต่อกับผู้ใช้ โมดูลการแปลงข้อความเป็นหน่วยเสียง โมดูลการสร้างรูปปาก และโมดูลการสร้างภาพเคลื่อนไหว สำหรับตัวแบบที่สามารถใช้โปรแกรมเสริมนี้ได้สามารถเป็นได้ทั้งกรณีตัวแบบที่มีการเตรียมรูปปากของแต่ละหน่วยเสียงเอาไว้แล้ว หรือกรณีตัวแบบธรรมดาที่ไม่มีการเตรียมรูปปากของแต่ละหน่วยเสียง ซึ่งโครงการนี้ได้นำเสนออัลกอริธึมเพื่อสร้างรูปปากแต่ละหน่วยเสียงอัตโนมัติเพื่อให้สามารถผสมรูปปากได้เช่นเดียวกับในกรณีแรกด้วย โดยในการพัฒนานี้จะใช้ Maya API ในการควบคุมโมเดลปากนี้ และผลที่ได้รับคือทำให้การสร้างภาพการเคลื่อนไหวของปากสำหรับภาษาไทยทำได้รวดเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	A Development of a Maya Plug-in for Lip Animation of Thai Language Speaking
Student	Mr. Pongrat Muangcharoen
Advisor	Assoc.Prof.Dr. Nopporn Chotikakamthorn
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2005

ABSTRACT

An important problem of Maya is that the process of creating lip animation is complex and takes a long time. Typically, users have to model their several lip models by themselves, then determine many syllable shapes from these lip models. Therefore this project offers a solution to solve the problem by a development of Maya plug-in. With this Maya plug-in, a user only loads his 3D model and types on his keyboard a paragraph of Thai text, and then the plug-in converts the paragraph of Thai text to a phoneme stream and then maps each phoneme to each 3D lip shape model. After that, the plug-in blends the phonemes together to a syllable and also blends the 3D lip shape models together to a syllable 3D lip shape model. Keyframe animation technique is used. This project consists of 5 modules : main module, user interface module, text-to-speech module, creating lip shape module, and animation module. The models used with this plug-in can be both models with all phoneme lip shapes and simple models. For the simple models, this project also offers an algorithm to create new phoneme lip shapes which will be used in blending to syllable lip shapes. In this development, Maya API will be used to control the lip models. The result is increasing the speed of creating lip animation of Thai language speaking.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้เขียนขอขอบพระคุณเป็นอย่างยิ่งสำหรับ รศ.ดร.นพพร โชติกกำธร ที่ได้ให้โอกาสและความเมตตา ขอบพระคุณคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชา ขอบพระคุณคุณพ่อ คุณแม่ พี่ชาย สำหรับการสนับสนุนในทุก ๆ ด้านและสุดท้ายขอขอบพระคุณ คุณเบญจมาศ จันทร์บุตรี สำหรับการสนับสนุนข้อมูลเกี่ยวกับหลักภาษาไทยเป็นอย่างดี

พงศ์รัฐ เมืองเจริญ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... I

บทคัดย่อภาษาอังกฤษ..... II

กิตติกรรมประกาศ..... III

สารบัญ..... IV

สารบัญตาราง..... VII

สารบัญรูป..... VIII

บทที่

1. บทนำ..... 1

1.1 โครงการพัฒนาระบบงาน..... 1

1.2 ประโยชน์ของโครงการ..... 2

1.3 สภาพแวดล้อมในการพัฒนา..... 2

1.4 แนวทางการพัฒนา..... 2

1.5 โครงสร้างเอกสาร..... 3

2. การออกแบบโปรแกรมเสริมมายาเพื่อสร้างภาพเคลื่อนไหวรูปปาก..... 5

2.1 โปรแกรมเสริมมายาเพื่อช่วยสร้างภาพการเคลื่อนไหวของปากในการพูดภาษาไทย 5

2.2 แนวคิด..... 5

2.3 โครงสร้างโปรแกรม..... 6

2.4 การออกแบบโปรแกรม..... 9

2.5 สรุป..... 14

3. โมดูลหลัก..... 15

3.1 โปรแกรมมายา..... 15

3.2 สถาปัตยกรรมมายา..... 16

3.3 แนวทางในการพัฒนาโปรแกรมเสริม..... 18

3.4 โปรแกรมเสริมของโครงการ..... 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.5 แนวคิดโปรแกรมเสริมของโครงการ..... 24
3.6 โครงสร้างข้อมูลและอัลกอริธึมโมดูลหลัก..... 25
3.7 สรุป..... 33
4. โมดูลย่อยส่วนติดต่อกับผู้ใช้..... 34
4.1 การสร้างโมดูลย่อยส่วนติดต่อกับผู้ใช้..... 34
4.2 โปรแกรมสร้างเมนูใหม่สำหรับโปรแกรมเสริม..... 35
4.3 โปรแกรมสร้างหน้าจอรับค่าข้อมูลใหม่สำหรับโปรแกรมเสริม..... 35
4.4 โปรแกรมการโหลดเสียง..... 40
4.5 การเก็บโปรแกรม..... 40
4.6 สรุป..... 40
5. โมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง..... 41
5.1 โปรแกรมวาจา (Vaja) : ซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทย..... 41
5.2 โครงสร้างของระบบสังเคราะห์เสียง..... 41
5.3 การวิเคราะห์ข้อความ (Text Analysis) ของ Vaja API..... 44
5.4 การสังเคราะห์เสียงพูด..... 48
5.5 การออกแบบ Class โมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง..... 50
5.6 โครงสร้างข้อมูลของไฟล์หน่วยเสียง..... 51
5.7 สรุป..... 51
6. โมดูลย่อยการสร้างรูปปากและโมดูลย่อยการสร้างภาพเคลื่อนไหว..... 52
6.1 การสร้างรูปปากสำหรับแต่ละหน่วยเสียงภาษาไทย..... 52
6.2 การสร้างรูปปากสำหรับแต่ละเสียงพยางค์..... 57
6.3 การสร้างส่วนควบคุมการผสมรูปปากด้วย Blend Shape Deformer..... 58
6.4 โมดูลย่อยการสร้างภาพเคลื่อนไหว..... 59
6.5 ขั้นตอนการทำงานของโมดูลย่อยสร้างการเคลื่อนไหว..... 59
6.6 สรุป..... 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

7. การสร้างรูปปากมาตรฐานอัตโนมัติ 62
7.1 การสร้างรูปปากมาตรฐานอัตโนมัติ 62
7.2 หลักการขยับปาก 62
7.3 วิธีการกำหนดข้อมูลเสียงสำหรับรูปปาก 64
7.4 ข้อมูลการกำหนดมุมและรูปปากใหม่ที่ได้ 64
7.5 การสร้างไฟล์เสียง 66
7.6 หลักการควบคุมการเคลื่อนไหวสำหรับโมเดล 3 มิติ 67
7.7 สรุป 67
8. ผลการศึกษาและพัฒนาโครงการ 68
8.1 โปรแกรมสร้างภาพการเคลื่อนไหวของปากในการพูดภาษาไทย 68
8.2 ขั้นตอนการใช้งาน 69
8.3 การสร้างภาพการเคลื่อนไหว 71
8.4 สรุป 72
9. บทสรุป 73
9.1 สรุป 73
9.2 ข้อจำกัดและปัญหาของโครงการ 73
9.3 ข้อเสนอแนะ 74
บรรณานุกรม 75
ภาคผนวก 76
ภาคผนวก ก. การแบ่งกลุ่ม Viseme เสียงในภาษาไทย 76
ภาคผนวก ข. คู่มือการติดตั้ง 78
ประวัติผู้เขียน 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 CRC ของ want2talkthai..... 10

2.2 CRC ของ CVajaTTS..... 11

2.3 CRC ของ CLipShape..... 11

2.4 CRC ของ ClipAnimation..... 12

3.1 รหัสควบคุมภายในไฟล์ w2tt.txt..... 29

4.1 รหัสควบคุมของโพรโตคอลการส่งข้อมูลเป็นไฟล์ w2tt.txt..... 36

4.2 ความหมายลำดับที่ตามหลังรหัสควบคุม –v..... 36

6.1 สัญลักษณ์หน่วยเสียงของพยัญชนะภาษาไทย..... 55

6.2 สัญลักษณ์หน่วยเสียงของพยัญชนะควบกล้ำภาษาไทย..... 56

6.3 สัญลักษณ์หน่วยเสียงของสระภาษาไทย..... 57

7.1 ข้อมูลการกำหนดค่ามุมรูปปาก..... 65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 โครงสร้างโปรแกรมเสริม 7
2.2 Collaboration Graphs ของโปรแกรมเสริม 13
3.1 โปรแกรมมายา 7 15
3.2 ระบบมายา 16
3.3 หน้าต่าง Hypergraph 17
3.4 กราฟดีเพนเดนซีของโพลีกอนรูปทรงกลม 18
3.5 ระบบมายาและโปรแกรมเสริมมายา 19
3.6 ลักษณะการทำงานของโปรแกรมเสริม 25
3.7 โครงสร้างข้อมูลบางส่วนของโมดูลหลัก 27
3.8 Flow chart อัลกอริธึมโมดูลหลัก 28
3.9 อัลกอริธึมในการแสดงส่วนติดต่อกับผู้ใช้ 29
3.10 ตัวอย่างข้อมูลไฟล์ w2tt.txt 30
3.11 อัลกอริธึมการกำหนดค่าโครงสร้างข้อมูล thaiPhonemes 31
3.12 อัลกอริธึมการโหลดไฟล์เสียงพูด 32
4.1 การสร้างโมดูลย่อยส่วนติดต่อกับผู้ใช้ 34
4.2 โปรแกรมสร้างเมนูใหม่ของโปรแกรมเสริม 35
4.3 โครงสร้างข้อมูลส่วนย่อยติดต่อกับผู้ใช้ 37
4.4 Flow chart อัลกอริธึมโปรแกรม want2talkthaiOptions.mel 38
4.5 อัลกอรีธึมการเรียกข้อมูลจากคอนโทรล 39
4.6 โปรแกรม want2talkthaiLoadAudioFile.mel 40
5.1 Block diagram ของโครงสร้างระบบสังเคราะห์เสียง 42
5.2 Block diagram ของการแยกประโยคภาษาไทย 45
5.3 โครงสร้างหน่วยครึ่งพยางค์ 49
5.4 แสดงคลาส CVajaTTS 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

5.5 ข้อมูลไฟล์หน่วยเสียงสำหรับข้อความ "คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง" 51
6.1 Flow chart อัลกอริธึมโมดูลย่อยการสร้างรูปปาก..... 54
6.2 Flow chart อัลกอริธึมการสร้างรูปปากเสียงพยางค์..... 58
6.3 Flow Chart อัลกอริธึมการค้นหาตำแหน่งและการกำหนด Keyframe..... 60
7.1 ตำแหน่งของจุด Control Vertex ทั้ง 4 ที่สัมพันธ์กับรูปปากหน่วยเสียง..... 63
7.2 ลักษณะการหมุนจุดรอบแกน X..... 63
7.3 ลักษณะการหมุนจุดรอบแกน Y..... 64
7.4 รูปปากที่ได้จากข้อมูลตารางที่ 7.1..... 66
8.1 หน้าต่างโปรแกรมเสริม..... 68
8.2 หน้าต่าง Plug-in Manager..... 69
8.3 เมนูเรียกใช้โปรแกรมเสริม..... 70
8.4 หน้าต่าง Render Setting..... 71
8.5 ภาพที่ได้จากการพูดคำว่า "สวัสดี"..... 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1. โครงการพัฒนาระบบงาน

ปัจจุบันคอมพิวเตอร์กราฟิกส์ 3 มิติ มีความสำคัญในชีวิตประจำวันของเรามากขึ้น เพราะสามารถทำให้เราได้เห็นภาพที่น่าตื่นตาตื่นใจ น่าหลงใหลและไม่น่าที่จะเป็นไปได้ต่างๆซึ่งเป็นการยากหรือเป็นไปไม่ได้เลยที่จะได้เห็นหากไม่ใช้คอมพิวเตอร์กราฟิกส์ 3 มิติ เช่น ในภาพยนตร์ที่ทำเงินมหาศาล เกมส์ที่ได้รับรางวัลชนะเลิศ สปอตโฆษณาทีวี การโฆษณาสิ่งพิมพ์ การทำบรรจุภัณฑ์สินค้า วิดีโอสื่อประสม และเว็บไซต์บริษัท เป็นต้น และสำหรับซอฟต์แวร์คอมพิวเตอร์กราฟิกส์ 3 มิติ เป็นที่นิยมใช้กันมากโปรแกรมหนึ่งก็คือ โปรแกรมมายา (Maya®) ของบริษัท Alias Systems จำกัด ซึ่งต่อไปขอเรียกว่า "มายา" ซึ่งเป็นซอฟต์แวร์ที่มีลักษณะสำคัญ (Features) หลายอย่างที่ช่วยในการสร้างภาพเคลื่อนไหว 3 มิติ โดยเวอร์ชั่นล่าสุดขณะนี้คือเวอร์ชั่น 7.0

โครงการนี้เป็นการศึกษา พัฒนา และประยุกต์ใช้งานโปรแกรมเสริมสำหรับมายา 7.0 (Plug-in for Maya®7) ที่ช่วยในการสร้างภาพการเคลื่อนไหวของปาก (Lip Animation) ในมายา ซึ่งลักษณะของโปรแกรมเสริมนี้จะเป็นฟังก์ชันใหม่ที่เพิ่มเข้าไปในมายา และใช้ในการกำหนดการเคลื่อนไหวของรูปปากของหน่วยเสียง (Visemes) ขณะพูดให้สัมพันธ์กับหน่วยเสียง (Phonemes) ต่างๆ ในช่วงเวลาความเร็วของการพูดที่กำหนดให้ เพื่อให้ปากสามารถขยับได้เหมือนกับพูดจริง โดยผู้ใช้จะต้องพิมพ์ข้อความที่ต้องการให้ปากพูดลงไปพร้อมทั้งกำหนดจังหวะความเร็วที่ต้องการให้พูด หลังจากนั้นโปรแกรมเสริมจะกำหนดค่าต่างๆ ที่ทำให้ปากสามารถขยับพูดได้โดยอัตโนมัติ พร้อมทั้งสามารถเปิดหรือปิดเสียงพูดสังเคราะห์ได้ สำหรับภาษาที่ใช้จะเป็นภาษาไทยและโปรแกรมเสริมนี้สามารถใช้ได้กับทุกโมเดล (Models) ที่มีการเตรียมสร้างภาพการเคลื่อนไหวของปากโดยใช้เทคนิค Blend Shape ดังนั้นการทำให้มนุษย์ สุนัข หนังสือหรืออะไรก็ตาม ให้สามารถพูดได้จึงไม่ใช่เรื่องยากอีกต่อไปในมายา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ประโยชน์ของโครงการ

โครงการนี้เป็นการศึกษาและการพัฒนาโปรแกรมทางคอมพิวเตอร์กราฟิกส์ 3 มิติ ซึ่งเป็นประโยชน์ต่อผู้ที่ใช้งานโปรแกรมมายาอยู่แล้วทั่วโลกที่ต้องการให้มีฟังก์ชันพิเศษเพิ่มเข้ามาในส่วนของการทำภาพการเคลื่อนไหวของปากตัวละครเป็นภาษาไทย ซึ่งจะทำให้ลดเวลาในการสร้างได้เป็นอย่างมาก

1.3 สภาพแวดล้อมในการพัฒนา

เครื่องมือหรืออุปกรณ์ที่ใช้ภายในโครงการนี้ประกอบด้วยเครื่องคอมพิวเตอร์ PC ซึ่งรัน (Run) โปรแกรมมายา เวอร์ชัน 7.0 สำหรับระบบปฏิบัติการ Windows XP และลำโพงสำหรับฟังเสียงพูด ในการพัฒนาโปรแกรมเสริมมายานี้ใช้ Microsoft Visual Studio.NET 2003 Visual C++ ในการคอมไพล์ (Compile) และลิงค์ (Link) ส่วน API ที่ใช้คือ Maya API สำหรับการสร้างโปรแกรมเสริมมายา ส่วนการปรับแต่ง User Interface ของมายาและโปรแกรมเสริมจะใช้ MEL ส่วนการสังเคราะห์เสียงพูดภาษาไทยและการแปลงข้อความภาษาไทยเป็นหน่วยเสียง จะใช้ Vaja API ของโปรแกรมวาจา (Vaja) เวอร์ชั่น 3.1 ซึ่งเป็นซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทยซึ่งถูกพัฒนาขึ้นโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)

1.4 แนวทางการพัฒนา

ในการพัฒนาโครงการนี้ ต้องใช้ความรู้ในหลายด้านและมีความเกี่ยวข้องกับเทคโนโลยีหลายแขนง อีกทั้งยังมีข้อจำกัดต่างๆที่ต้องคำนึง โดยสรุปเป็นประเด็นสำคัญ คือ

1) การพัฒนาโปรแกรมที่มีลักษณะเป็นโปรแกรมเสริมสำหรับมายา (Maya Plug-in)

2) การพัฒนาโปรแกรมที่ใช้สำหรับช่วยในการสร้างภาพเคลื่อนไหวของปากในมายา

3) การพัฒนาโปรแกรมที่ทำสำหรับการพูดภาษาไทย

1.4.1 การพัฒนาโปรแกรมที่มีลักษณะเป็นโปรแกรมเสริมสำหรับมายา (Maya Plug-in)

เนื่องจากมายาเป็นโปรแกรมสำหรับการสร้างภาพเคลื่อนไหว 3 มิติระดับโลกที่เป็นที่นิยมใช้กันมากในหลายๆ งานเกี่ยวกับคอมพิวเตอร์กราฟิกส์ และมีผู้ใช้โปรแกรมนี้จำนวนมากทั่วโลก ดังนั้นการพัฒนาโปรแกรมเสริมที่จะเพิ่มความสามารถให้กับมายาในการสร้างภาพการเคลื่อนไหวของปากจึงเป็นสิ่งที่จะช่วยอำนวยความสะดวกแก่ผู้ใช้จำนวนมากในการจะพัฒนาระบบการสร้างภาพการเคลื่อนไหวของปาก ซึ่งโปรแกรมเสริมนี้สามารถนำไปใช้งานได้กับทุกโมเดลที่ต้องการทำเป็นภาพเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.2 การพัฒนาโปรแกรมที่ใช้สำหรับช่วยในการสร้างภาพเคลื่อนไหวของปากในมายา

ในมายาไม่มีฟังก์ชัน (Function) เฉพาะสำหรับช่วยอำนวยความสะดวกในการสร้างภาพการเคลื่อนไหวของปาก ดังนั้นในการสร้างภาพการเคลื่อนไหวของปากนี้ด้วยมายาแบบปกติ ผู้ใช้จึงต้องเสียเวลาพัฒนาใน 2 ส่วน ส่วนแรกคือส่วนการสร้างโมเดลปาก (Section of Modeling) และส่วนที่สองคือส่วนการทำให้โมเดลปากนี้เคลื่อนไหว (Section of Animation) ด้วยการใส่คีย์เฟรม (Keyframe) ในไทม์ไลน์ (Time Line) ของมายาด้วยตัวเองซึ่งต้องใช้เวลานานและอาจได้ภาพการเคลื่อนไหวที่ไม่ตรงกับเสียงพูดทำให้ไม่สมจริง การพัฒนาโครงการนี้จะช่วยเพิ่มความเร็วและเพิ่มความสมจริงในการสร้างภาพการเคลื่อนไหวของปาก โดยทำงานในส่วนที่สองคือจัดการในส่วนของการทำให้โมเดลปากพร้อมที่จะเคลื่อนไหว ด้วยการวิเคราะห์หน่วยเสียงจากข้อความและใส่คีย์เฟรมในไทม์ไลน์ของมายาให้อย่างอัตโนมัติ

1.4.3 การพัฒนาโปรแกรมที่ทำสำหรับการพูดภาษาไทย

ในแต่ละระบบภาษาที่ต่างกันก็จะมีการแบ่งหน่วยเสียงแตกต่างกันไปตามภาษานั้นๆ เช่น ในระบบภาษาอังกฤษก็จะมีจำนวนหน่วยเสียงไม่เท่ากันและไม่เหมือนกับในระบบภาษาไทย เนื่องจากโปรแกรมเสริมนี้พัฒนาขึ้นเพื่อใช้งานกับภาษาไทย ดังนั้นหากจะนำไปใช้กับภาษาอื่นๆ คุณภาพของทั้งในด้านรูปปากที่มีให้เห็นหรือเสียงพูดที่เปล่งออกมาจะไม่ดีเหมือนกับในภาษาไทย

1.5 โครงสร้างเอกสาร

เอกสารโครงการนี้ประกอบด้วย 9 บท คือ

บทที่ 1) บทนำ

บทที่ 2) การออกแบบโปรแกรมเสริมมายาเพื่อสร้างภาพเคลื่อนไหวรูปปาก

บทที่ 3) โมดูลหลัก

บทที่ 4) โมดูลย่อยส่วนติดต่อกับผู้ใช้

บทที่ 5) โมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง

บทที่ 6) โมดูลย่อยการสร้างรูปปากและโมดูลย่อยการสร้างภาพเคลื่อนไหว

บทที่ 7) การสร้างรูปปากมาตรฐานอัตโนมัติ

บทที่ 8) ผลการศึกษาและพัฒนาโครงการ

บทที่ 9) บทสรุป

โดยบทที่ 2 กล่าวถึงการออกแบบโดยรวมของโปรแกรมเสริม Logic ของโปรแกรมเสริม ทฤษฎีและเหตุผลของการออกแบบดังกล่าว ส่วนบทที่ 3 กล่าวถึงการพัฒนาโปรแกรมเสริมสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมมายาและโมดูลหลัก ส่วนบทที่ 4 กล่าวถึงการพัฒนาส่วนติดต่อกับผู้ใช้ รวมถึงโปรแกรมต่างๆที่เกี่ยวข้อง ส่วนบทที่ 5 กล่าวถึงการพัฒนาโมดูลย่อยการแปลงข้อความเป็นหน่วยเสียงโดยใช้ไลบรารีจากโปรแกรม Vaja ส่วนบทที่ 6 กล่าวถึงการพัฒนาโมดูลย่อยการสร้างรูปปาก ส่วนบทที่ 7 กล่าวถึงการพัฒนาโมดูลย่อยการสร้างภาพเคลื่อนไหวให้ปากมีการเคลื่อนไหวไปด้วยกันกับเสียงพูด ส่วนบทที่ 8 เป็นผลที่ได้จากการพัฒนาโครงการและวิธีใช้งานโปรแกรมเสริม และบทที่ 9 เป็นการสรุปผลที่ได้ ปัญหา และแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การออกแบบโปรแกรมเสริมมายาเพื่อสร้างภาพเคลื่อนไหวรูปปาก

2.1 โปรแกรมเสริมมายาเพื่อช่วยสร้างภาพการเคลื่อนไหวของปากในการพูดภาษาไทย

โดยปกติแล้ว วิธีที่นิยมใช้กันในการสร้างภาพการเคลื่อนไหวของปากให้ตัวละครในโปรแกรมมายานั้น จะทำโดยการใช้เทคนิคของ Blend Shape ซึ่งประกอบด้วย Based Shape และ Target Shapes ต่างๆ ซึ่ง Based Shape จะเป็นรูปปากปกติของตัวละคร ส่วน Target Shapes ต่างๆ จะเป็นรูปปากของหน่วยเสียงต่างๆ เช่น รูปปากวงกลมของหน่วยเสียง "อู" เป็นต้น นั่นคือผู้ใช้จะต้องสร้าง Target Shapes ต่างๆ ของรูปปากในการออกเสียงหน่วยเสียงต่างๆ ขึ้นมาด้วยตัวเอง เมื่อสร้าง Target Shapes ต่างๆ ได้แล้วจึงมาปรับแต่งตำแหน่งคีย์เฟรมที่เวลาต่างๆ ให้ปากของตัวละครมีการขยับเหมือนกับการพูด ซึ่งทุกขั้นตอนการสร้างนั้นเป็นงานที่ต้องใช้ความอดทนและความพยายามอย่างสูง

โปรแกรมเสริมนี้จะช่วยให้การสร้างภาพเคลื่อนไหวของปากให้มีความรวดเร็วและแน่นอนมากขึ้น โดยจะสร้าง Target Shapes ต่างๆ ของรูปปากหน่วยเสียงให้อัตโนมัติให้แก่ตัวละคร ส่วนการกำหนดคีย์เฟรม ผู้ใช้เพียงแค่กำหนดจังหวะช้าหรือเร็วของการพูดในแต่ละพยางค์ให้กับโปรแกรมเสริมนี้เท่านั้น ส่วนที่เหลือทั้งหมด เช่น การปรับค่าน้ำหนักของแต่ละ Blend Shape นั้น โปรแกรมเสริมนี้จะทำให้โดยอัตโนมัติ หลังจากนั้นผู้ใช้สามารถ Playback ฉากภาพเคลื่อนไหวนั้นได้ทันที

2.2 แนวคิด

ในการพัฒนาโปรแกรมเสริมมายานี้จะเกี่ยวข้องกับกระบวนการหลายอย่าง คือ การสร้างโปรแกรมเสริมมายา การสร้างส่วนติดต่อกับผู้ใช้ การแปลงข้อความเป็นหน่วยเสียงและเสียงสังเคราะห์ การสร้างรูปปาก Target ของแต่ละหน่วยเสียง และ การปรับค่าน้ำหนักของ Blend Shape และกำหนดคีย์เฟรม

Maya API เป็น API ที่โปรแกรมมายาเตรียมไว้ให้ผู้ใช้ที่ต้องการพัฒนาโปรแกรมเสริมขึ้นใช้เอง โดย Maya API นี้จะใช้ได้กับ Microsoft Visual Studio.NET 2003 Visual C++ ดังนั้นจึงสามารถนำมาใช้ในส่วนของการสร้างโปรแกรมเสริมมายา

MEL เป็นภาษาสคริปต์ของมายาที่สามารถใช้สร้างส่วนติดต่อกับผู้ใช้ได้

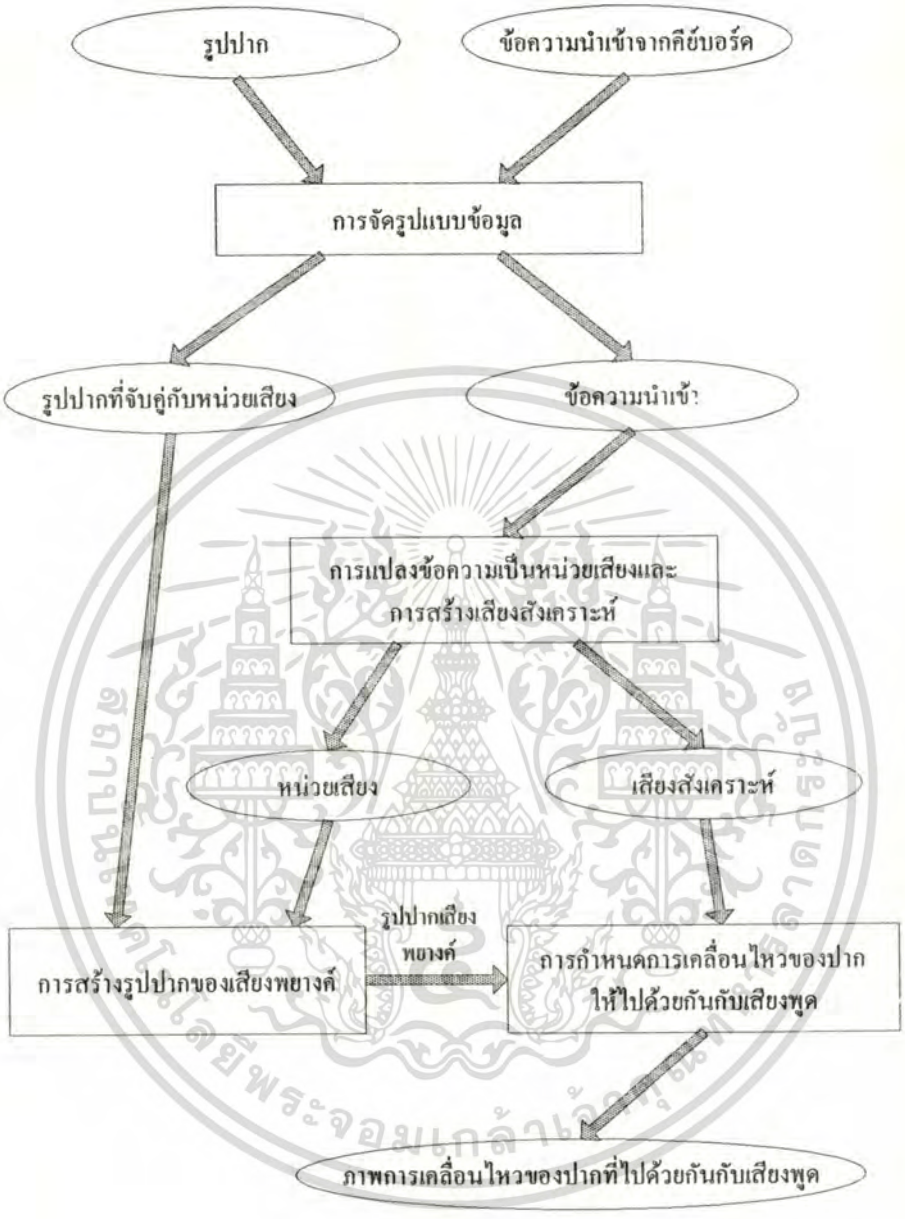
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vaja API เป็น API ที่ถูกพัฒนาโดย NECTEC สำหรับใช้ในการแปลงข้อความเป็นหน่วยเสียงและเสียงสังเคราะห์

2.3 โครงสร้างโปรแกรมเสริม

โปรแกรมเสริมนี้ทำงานภายใต้ระบบของโปรแกรมมายา บนระบบปฏิบัติการ Windows XP มีการเรียกใช้ไลบรารี Vaja API ของโปรแกรม Vaja ซึ่งอาศัยความสามารถของโปรแกรม Vaja ในการแปลงข้อความนำเข้าเป็นไฟล์เสียงและไฟล์หน่วยเสียง เพื่อนำมาใช้ในกระบวนการกำหนดการเคลื่อนไหวของปาก โครงสร้างของโปรแกรมเสริมแสดงได้ดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 โครงสร้างโปรแกรมเสริม

เมื่อผู้ใช้รัน (Run) โปรแกรมมายาและเปิดไฟล์ของตัวแบบที่ต้องการทำภาพการเคลื่อนไหวของปากแล้ว ขั้นตอนต่อไป คือ การพิมพ์ข้อความเข้าสู่ระบบ จากนั้นข้อความนำเข้าเหล่านี้จะถูกแปลงเป็นหน่วยเสียงและสร้างเป็นเสียงพูดสังเคราะห์โดยขั้นตอนการแปลงข้อความเป็นหน่วยเสียงและการสร้างเสียงสังเคราะห์ ผลที่ได้จากขั้นตอนนี้มี 2 อย่าง คือ ไฟล์หน่วยเสียงที่แสดงลักษณะของเสียงที่เกิดจากข้อความนำเข้าที่ถูกพิมพ์เข้ามา โดยไฟล์หน่วยเสียงนี้เป็นชนิดข้อความ (Text file)และไฟล์เสียงพูดสังเคราะห์ซึ่งเกิดจากข้อความนำเข้าที่ถูกพิมพ์เข้ามานี้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยไฟล์เสียงนี้เป็นชนิด .wav ขั้นตอนต่อไปคือการสร้างรูปปากของเสียง โดยนำข้อมูลหน่วยเสียงจากไฟล์หน่วยเสียงมาจับคู่กับรูปปากของแต่ละหน่วยเสียงแล้วผสมกันเป็นรูปปากของเสียงที่เป็นพยางค์ ผลที่ได้จากขั้นตอนนี้คือรูปปากที่แสดงการพูดหนึ่งพยางค์ จากนั้นขั้นตอนต่อไปคือการกำหนดการเคลื่อนไหวของปากให้ไปด้วยกันกับเสียงพูด เมื่อเราได้รูปปากของพยางค์ที่ต้องการแล้วจึงนำรูปปากนี้มากำหนดเวลาที่จะแสดงผลรูปปากนี้เพื่อให้เข้ากับเสียงที่ได้จากการสังเคราะห์ ผลที่ได้คือไฟล์ภาพการเคลื่อนไหวของปากที่ไปด้วยกันกับเสียงพูด ซึ่งเป็นชนิด .avi

ขั้นตอนการจัดรูปแบบข้อมูลเป็นการรับข้อความจากผู้ใช้ซึ่งข้อความนี้จะถูกพิมพ์เข้ามาโดยผู้ใช้ ดังนั้นจึงต้องสร้างส่วนติดต่อกับผู้ใช้ขึ้นมาใหม่ให้กับโปรแกรมมายา ซึ่งพัฒนาโดยใช้ภาษา MEL (Maya Embedded Language) ซึ่งเป็นภาษาสคริปต์ (Script language) ของโปรแกรมมายาเอง นอกจากจะรับข้อความจากผู้ใช้แล้วในขั้นตอนนี้จะให้ผู้ใช้จับคู่รูปปากเข้ากับหน่วยเสียงภาษาไทย ข้อความที่ผู้ใช้พิมพ์เข้ามาและข้อมูลการจับคู่รูปปากจะถูกบันทึกเป็นไฟล์ชนิดข้อความเก็บไว้ที่ฮาร์ดดิสก์ (Harddisk)

ขั้นตอนการแปลงข้อความเป็นหน่วยเสียงและการสร้างเสียงสังเคราะห์จะอ่านข้อความนำเข้าจากไฟล์มาเก็บในบัฟเฟอร์ (Buffer) แล้วนำข้อความจากบัฟเฟอร์ไปแปลงให้เป็นไฟล์ข้อมูลหน่วยเสียงและไฟล์เสียงพูดของข้อความนั้น โดยการประมวลผลในขั้นตอนนี้จะใช้ฟังก์ชันของไลบรารีจากโปรแกรมวาจา ซึ่งเป็นโปรแกรมประเภทแปลงข้อความเป็นคำพูด (Text-to-speech) ซึ่งเป็นผลงานวิจัยและพัฒนาของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ดังนั้นระบบนี้จึงจำเป็นต้องติดตั้งโปรแกรมวาจาลงในเครื่องด้วยเพื่อให้สามารถใช้งานไลบรารีนี้ได้

ขั้นตอนการสร้างรูปปากของเสียงพยางค์ทำหน้าที่ในการผสมรูปปากหน่วยเสียงต่างๆที่ประกอบกันเป็นพยางค์ให้กลายเป็นรูปปากใหม่ โดยเทคนิคที่ใช้ในการผสมรูปปากนี้เรียกว่า Blend Shape ซึ่งเป็นเทคนิคในการปรับเปลี่ยนรูปทรงของโปรแกรมมายา

เมื่อได้รูปปากของเสียงพยางค์ที่ต้องการแล้ว ขั้นตอนต่อไปคือการกำหนดการเคลื่อนไหวของปากให้ไปด้วยกันกับเสียงพูดโดยการนำรูปปากและเสียงสังเคราะห์ที่ได้มากำหนดจังหวะการเคลื่อนไหวเพื่อให้การขยับปากไปพร้อมกับเสียงพูดสังเคราะห์ โดยเทคนิคที่เกี่ยวข้องในการกำหนดนี้คือ Keyframe ซึ่งเป็นเทคนิคที่นิยมใช้ควบคุมการเคลื่อนไหว เมื่อกำหนดการเคลื่อนไหวเรียบร้อยแล้วจะได้การเคลื่อนไหวที่สามารถนำไปสร้างเป็นไฟล์ .avi ได้ตามวิธีการของโปรแกรมมายาตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การออกแบบโปรแกรมเสริม

โดย CRC เป็นการออกแบบ Classes หรือ Objects ที่มีในระบบว่าเป็น Class อะไร สืบทอดมาจาก Class อะไร มีหน้าที่อะไรบ้าง และในหน้าที่นั้นๆ ต้องทำงานร่วมกับ Classes หรือ Objects ใดบ้าง ซึ่งชื่อ Class จะอยู่ด้านบนซ้ายของตาราง ตามด้วยเครื่องหมาย Colon และชื่อ Class ที่สืบทอดมา หากมีการการสืบทอด ส่วนด้านบนขวาจะแสดงประเภทของ Class ว่าเป็นชนิด Abstract หรือ Concrete และเป็น Subsystem หรือไม่ Abstract Class คือ Base Class ซึ่งคลาสอื่นสืบทอดมา ส่วน Concrete Class คลาสซึ่งสืบทอดมาจาก Base Class และใช้ทุกฟังก์ชันใน Base Class โดยภายในตารางแยกเป็นด้านซ้ายแสดงหน้าที่ของ Class นั้น ซึ่งถูกกำกับด้วยหมายเลข ส่วนด้านขวาเป็น Classes ที่ทำงานร่วมเพื่อให้หน้าที่ในข้อนั้นๆสำเร็จ

ในส่วน Collaboration Graphs คือการสร้าง Diagram เพื่อให้เห็นความสัมพันธ์ของ Class ที่มีในร ะบบให้เข้าใจง่ายขึ้น โดยแสดงเป็นรูปสี่เหลี่ยมที่แสดงชื่อ Class ส่วนที่ของของสี่เหลี่ยมแสดงหมายเลขซึ่งตรงกับหมายเลขหน้าที่ใน CRC ของ Class นั้นๆ แล้วจะเชื่อมโยงด้วยลูกศรไปยังหมายเลขหน้าที่ของ Class ที่ทำงานร่วมกันเพื่อให้หน้าที่นั้นสำเร็จ โดยที่ทิศทางของหัวลูกศรจะหมายถึงการใช้บริการจากหน้าที่ข้อที่ชี้ของ Class ที่ถูกชี้อยู่

2.4.1 CRC

โมดูลหลัก ประกอบด้วย 4 โมดูลย่อย คือ โมดูลย่อยการติดต่อกับผู้ใช้ โมดูลย่อยแปลงข้อความเป็นไฟล์หน่วยเสียงและไฟล์เสียง โมดูลย่อยการสร้างรูปปาก และโมดูลย่อยการกำหนดค่าการทำภาพเคลื่อนไหว โมดูลย่อยการติดต่อกับผู้ใช้นั้นเป็นโมดูลเดียวที่อยู่ภายนอกระบบโปรแกรมเสริม ซึ่งมีลักษณะเป็นโปรแกรมเสริมแบบ MEL script ข้อมูลผลลัพธ์ (Output data) จากโมดูลย่อยการติดต่อกับผู้ใช้จะถูกบันทึกเป็นไฟล์ข้อความเพื่อให้โปรแกรมเสริมหลักดึงไปประมวลผลต่อไป ดังนั้น Class ที่จะวิเคราะห์ CRC จึงเหลือเพียง 4 Classes ซึ่งอยู่ในโมดูลหลัก คือ want2talkthai CVajaTTS CLipShape และ CLipAnimation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 CRC ของ want2talkthai

want2talkthai: MpxCommand	Concrete
1. Display the plug-in user interface	
2. Set input data into the plug-in data structure	
3. Convert the input text to a phoneme file	CVajaTTS
4. Convert the input text to a speech sound file	CVajaTTS
5. Load the speech sound file into time line	
6. Create lip shapes for each syllable sound	CLipShape
7. Create phoneme lip shapes	CLipShape
8. Create blend shape deformers for each phoneme lip shape	CLipShape
9. Find the positions of keyframes to make synchronization between lip shape and speech	CLipAnimation
10. Set keyframes to synchronize between the lip animation and the speech sound	CLipAnimation

want2talkthai class จะเป็นส่วนหลักของระบบโปรแกรมเสริมซึ่งรับผิดชอบในการควบคุมขั้นตอนการทำงานโดยรวมของโมดูลย่อยโปรแกรมเสริมอื่นๆ Class นี้มีหน้าที่ 1) แสดงส่วนติดต่อผู้ใช้เพื่อรับค่าที่ต้องการเข้าสู่โปรแกรมเสริม 2) ปรับค่าที่ได้จากผู้ใช้เข้ากับโครงสร้างข้อมูลของโปรแกรมเสริม 3) แปลงข้อความเป็นไฟล์หน่วยเสียง 4) แปลงข้อความเป็นไฟล์เสียงพูด 5) โหลดเสียงพูดเข้าสู่ Scene ของโปรแกรมมายา 6) สร้างรูปปากของเสียงพยางค์ 7) สร้างรูปปากของแต่ละหน่วยเสียง 8) สร้าง Blend shape deformer สำหรับควบคุมรูปปากของแต่ละหน่วยเสียง 9) คำนวณตำแหน่งการวาง Keyframe เพื่อการทำภาพเคลื่อนไหวของปาก 10) กำหนดการเคลื่อนไหวของปากให้ไปด้วยกันกับเสียงพูดโดยการกำหนด Keyframe ส่วน Collaborators คือ CVajaTTS ใช้สำหรับการแปลงข้อความเป็นไฟล์หน่วยเสียงและไฟล์เสียงพูด CLipShape ใช้สำหรับการสร้างรูปปากของเสียงพยางค์ และ CLipAnimation ใช้สำหรับการกำหนดการเคลื่อนไหวของปากให้ไปด้วยกันกับเสียงพูด

MPxCommand เป็น Class ในไลบรารี OpenMaya ของโปรแกรมมายา สำหรับการใส่คำสั่งที่ต้องการให้ทำงานในโปรแกรมเสริม Class นี้จะต้องมี doIt method ซึ่งเป็น Method ที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานเมื่อมีการเรียกใช้โปรแกรมเสริม หน้าที่ของ MPxCommand คือ 1) เก็บคำสั่งที่จะใช้ทำงานของโปรแกรมเสริม โดยมี Collaborator คือ MArgList class ซึ่งเป็น Class ที่ใช้เพื่อสร้างและดึงรายการ Argument lists ซึ่งสามารถถูกส่งไปให้กับ Methods doIt สำหรับรายละเอียดของ MArgList class จะไม่ขอกล่าวถึง

ตารางที่ 2.2 CRC ของ CVajaTTS

CVajaTTS	Abstract Subsystem
1. Convert a text string to a phoneme file 2. Convert a text string to a speech sound file	

CVajaTTS class เป็น class ที่ใช้ในการแปลงข้อความให้เป็นไฟล์หน่วยเสียงและไฟล์เสียงพูด .wav ซึ่งใช้ไลบรารี Vaja API ของโปรแกรม Vaja ซึ่งเป็นซอฟต์แวร์ที่ใช้ในการสังเคราะห์เสียงพูดภาษาไทย CVajaTTS class มีหน้าที่ 1) แปลงข้อความให้เป็นไฟล์หน่วยเสียง 2) แปลงข้อความให้เป็นไฟล์เสียงพูด

ตารางที่ 2.3 CRC ของ CLipShape

CLipShape	Abstract Subsystem
1. Create sylable lip shapes 2. Create phoneme lip shapes 3. Create blend shape deformers for each phoneme lip shape	

CLipShape เป็น Class ที่ใช้สำหรับการสร้างรูปปากของเสียงพยางค์ โดยรูปปากของเสียงพยางค์หนึ่งๆนั้นจะเกิดจากการผสมกันของรูปปากของหน่วยเสียงหลายหน่วยเสียง CLipShape class มีหน้าที่ 1) สร้างรูปปากของเสียงพยางค์ 2) สร้างรูปปากของหน่วยเสียงต่างๆ 3) สร้าง Blend shape deformer สำหรับแต่ละรูปปากของหน่วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

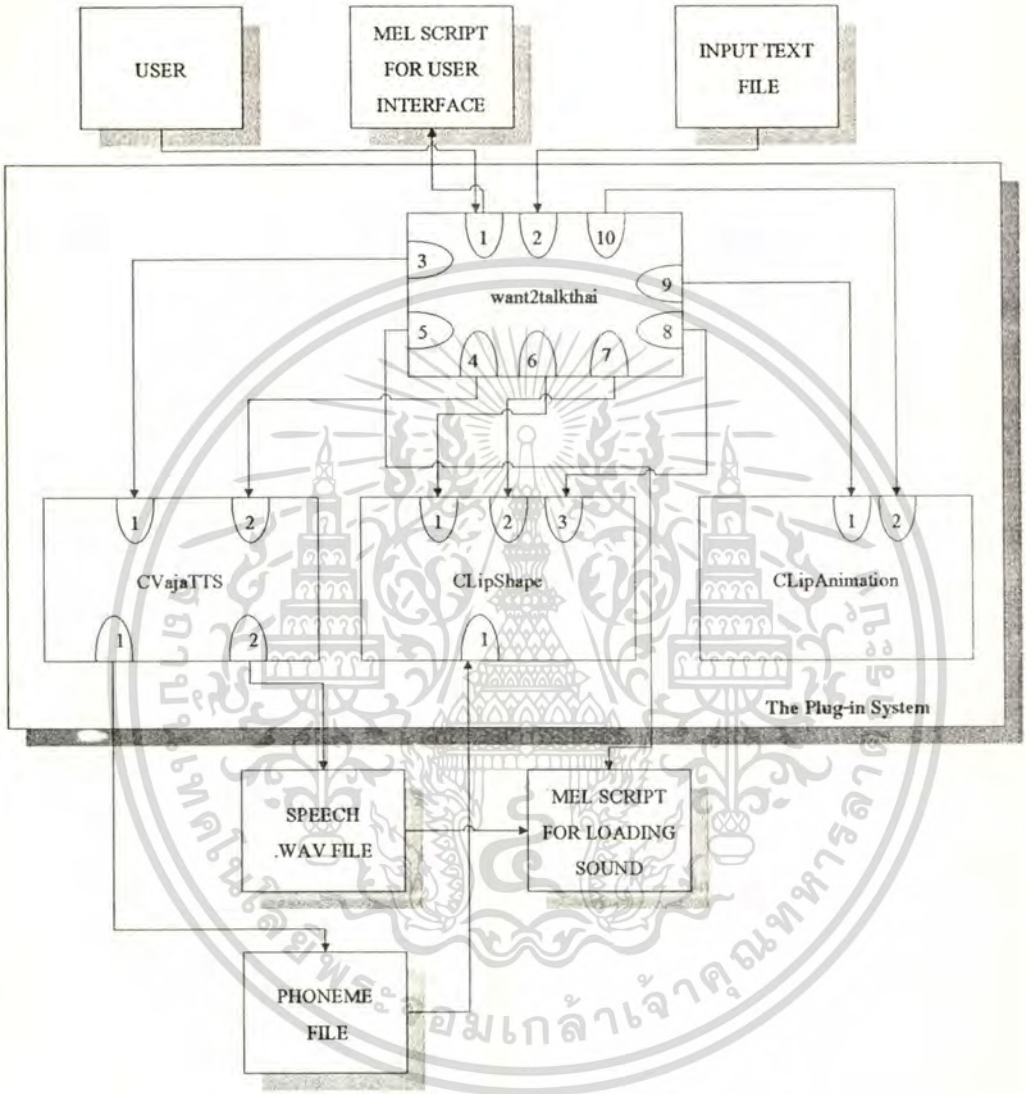
ตารางที่ 2.4 CRC ของ CLipAnimation

ClipAnimation	Abstract Subsystem
1. Find the positions of keyframes to make synchronization between lip shape and speech 2. Set keyframes for lip animation	

CLipAnimation class ใช้สำหรับการกำหนดจุด Keyframe ใน Time line สำหรับการทำภาพเคลื่อนไหวของปากให้ไปด้วยกันกับเสียงพูด Class นี้มีหน้าที่ 1) หาตำแหน่งเวลาที่จะกำหนดจุด Keyframe 2) กำหนดจุด Keyframe สำหรับควบคุมการเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 Collaboration Graphs



รูปที่ 2.2 Collaboration Graphs ของโปรแกรมเสริม

จากรูปที่ 2.2 แสดงถึงความสัมพันธ์ของ Classes ต่างๆ หรือโมดูลย่อยต่างๆ ภายในระบบโปรแกรมเสริม โดยความสัมพันธ์แสดงถึงการให้บริการในหน้าที่และการใช้บริการในหน้าที่ต่างๆ ของแต่ละ Class เพื่อให้หน้าที่นั้นๆ สำเร็จได้ เริ่มจาก want2talkthai class หน้าที่ 1) แสดงส่วนติดต่อผู้ใช้โดยการเรียกใช้บริการโปรแกรม MEL script เพื่อรับค่าที่ต้องการเข้าสู่โปรแกรมเสริม 2) ปรับค่าที่ได้จากผู้ใช้เข้ากับโครงสร้างข้อมูลของโปรแกรมเสริมโดยการอ่านข้อความและข้อมูลโมเดลจากไฟล์ข้อความนำเข้า 3) แปลงข้อความเป็นไฟล์หน่วยเสียงโดยการขอใช้บริการที่ 1 ของ CVajaTTS class 4) แปลงข้อความเป็นไฟล์เสียงพูด โดยการขอใช้บริการที่ 2 ของ CVajaTTS class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) โหลดเสียงพูดเข้าสู่ Scene ของโปรแกรมมายา 6) สร้างรูปปากของเสียงพยางค์ โดยการขอใช้บริการที่ 1 ของ CLipShape class 7) สร้างรูปปากของหน่วยเสียงแต่ละเสียง โดยการขอใช้บริการที่ 2 ของ CLipShape class 8) สร้าง Blend shape deformer เพื่อใช้สำหรับการเปลี่ยนรูปปาก โดยการขอใช้บริการที่ 3 ของ CLipShape class 9) คำนวณหาตำแหน่งการวาง Keyframe เพื่อการทำภาพเคลื่อนไหว โดยการขอใช้บริการที่ 1 ของ CLipAnimation class 10) กำหนดการเคลื่อนไหวของปากให้ไปด้วยกันกับเสียงพูด โดยการกำหนด Keyframe และขอใช้บริการที่ 2 ของ CLipAnimation class

CVajaTTS class หน้าที่ 1) แปลงข้อความให้เป็นไฟล์หน่วยเสียงแสดงการบันทึกลงไฟล์หน่วยเสียง (Phoneme file) 2) แปลงข้อความให้เป็นไฟล์เสียงพูด แสดงการบันทึกลงไฟล์เสียงพูด (Speech .wav file)

CLipShape class หน้าที่ 1) สร้างรูปปากของเสียงพยางค์ เพื่อให้บรรลุวัตถุประสงค์จะมีการอ่านข้อมูลหน่วยเสียงจากไฟล์หน่วยเสียงด้วย 2) สร้างรูปปากของหน่วยเสียงต่างๆ สามารถทำได้โดยไม่ต้องใช้บริการ Class ใด 3) สร้าง Blend shape deformer สำหรับแต่ละรูปปากของหน่วยเสียง สามารถทำได้โดยไม่ต้องใช้บริการ Class ใด

CLipAnimation class หน้าที่ 1) หาตำแหน่งเวลาที่จะกำหนดจุด Keyframe 2) กำหนดจุด Keyframe สำหรับควบคุมการเคลื่อนไหว

2.5 สรุป

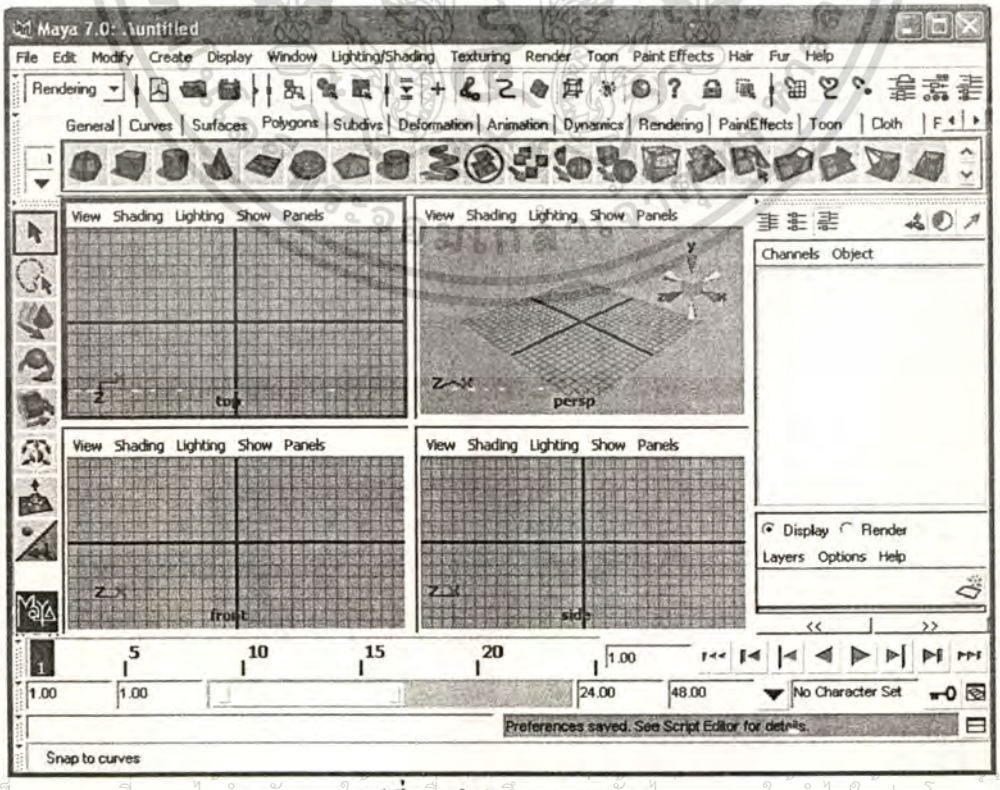
ในการออกแบบระบบโปรแกรมเสริมจะแบ่งเป็น 4 โมดูลย่อยซึ่งทำหน้าที่ต่างกันคือ 1) โมดูลย่อยส่วนติดต่อกับผู้ใช้ 2) โมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง 3) โมดูลย่อยการสร้างรูปปาก 4) โมดูลย่อยการสร้างภาพเคลื่อนไหวของปาก การออกแบบให้โมดูลย่อยส่วนติดต่อกับผู้ใช้เป็นโปรแกรมเสริมแบบ MEL script เพราะทำได้ง่าย แต่ก็มีปัญหาในการสื่อสารข้อมูลกับระบบโปรแกรมหลักคือการส่งค่าข้อความจะมีจำนวนจำกัด ดังนั้นจึงแก้ปัญหาด้วยการใช้ไฟล์ข้อความในการส่งและรับข้อมูลระหว่างโมดูลย่อยส่วนติดต่อกับผู้ใช้และระบบโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โมดูลหลัก

3. 1 โปรแกรมมายา

มายาเวอร์ชั่น 7 (Maya 7) มีอยู่ด้วยกัน 3 เอดิชัน (Edition) ซึ่งกล่าวเรียงลำดับจากเอดิชันใหญ่มาเอดิชันเล็กได้คือ มายา อันลิมิเต็ด (Maya Unlimited) มายา คอมพลีต (Maya Complete) และมายา เพอซอนแนล เลิร์นนิ่ง (Maya Personal Learning) สำหรับในรายงานนี้จะใช้มายาเวอร์ชั่น 7 คอมพลีต ในการอ้างอิง เนื่องจากเป็นเอดิชันที่ไม่ใหญ่มากและมีฟังก์ชัน (Function) การทำงานที่สมบูรณ์เพียงพอต่อการเขียนโปรแกรมเสริมตามจุดประสงค์ของรายงาน โดยในมายาเอดิชันนี้จะมีทูล (Tool) สำหรับนักเขียนโปรแกรม (Programer) เพื่อให้สามารถพัฒนาโปรแกรมเสริมใหม่ๆ เพิ่มเข้าไปสู่ตัวระบบซอฟต์แวร์เองได้ และทูลที่จะกล่าวถึงนี้ก็คือ Maya API (Maya Application Programmer Interface) นั่นเอง สำหรับรายละเอียดของโปรแกรมมายา 7 สามารถดูได้จากเว็บไซต์ http://www.alias.com ส่วนหน้าจอโปรแกรมแสดงดังรูปที่ 3.1

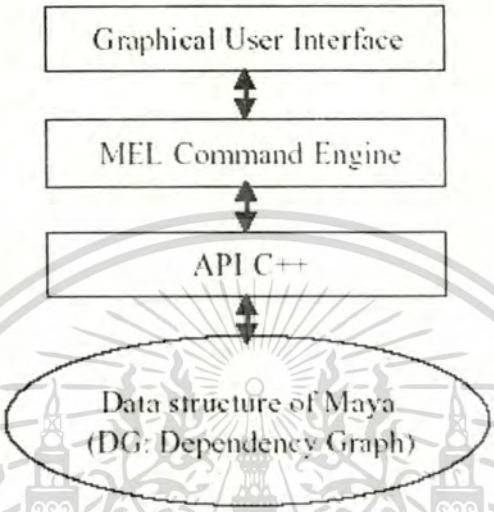


รูปที่ 3.1 โปรแกรมมายา 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 สถาปัตยกรรมมายา (Maya Architecture)

โครงสร้างของมายาแบ่งได้เป็น 4 ระดับ (Level) ดังรูปที่ 3.2



รูปที่ 3.2 ระบบมายา

จากรูปที่ 3.2 ระดับล่างสุดเป็นฐานข้อมูลที่เก็บข้อมูลด้านกราฟิกส์ และเรียกฐานข้อมูลนี้ว่าดีเพนเดนซีกราฟ หรือ ดีจี (DG : Dependency Graph) และข้อมูลในดีจี (DG) จะถูกเก็บในวัตถุ (Object) ที่เรียกว่าโหนด (Node) และแต่ละโหนดมีพร็อพเพอร์ตี (Property) ที่เรียกว่าแอตทริบิวต์ (Attribute) ซึ่งเก็บค่าคุณลักษณะ (Characteristics) ของแต่ละโหนด และแอตทริบิวต์เป็นตัวเชื่อมให้แต่ละโหนดสามารถเชื่อมต่อ (Connect) กันและส่งค่าข้อมูลให้กันได้

ระดับที่ใกล้ชิดกับดีจี (DG) ที่สุดคือระดับ API C++ ซึ่งหมายถึง Maya API ดังนั้น Maya API จึงทำงานเร็วที่สุด

ระดับชั้น MEL Command Engine จะรองรับการทำงานของ Graphical User Interface (GUI) โดยทุกคำสั่งจาก GUI จะต้องดำเนินการด้วย MEL Command Engine

ระดับชั้น Graphical User Interface เป็นส่วนที่ติดต่อกับผู้ใช้ของมายา โดยอำนวยความสะดวกให้ผู้ใช้มากที่สุดแต่ก็ทำงานช้าที่สุดด้วย

3.2.1 กราฟดีเพนเดนซี (Dependency Graph)

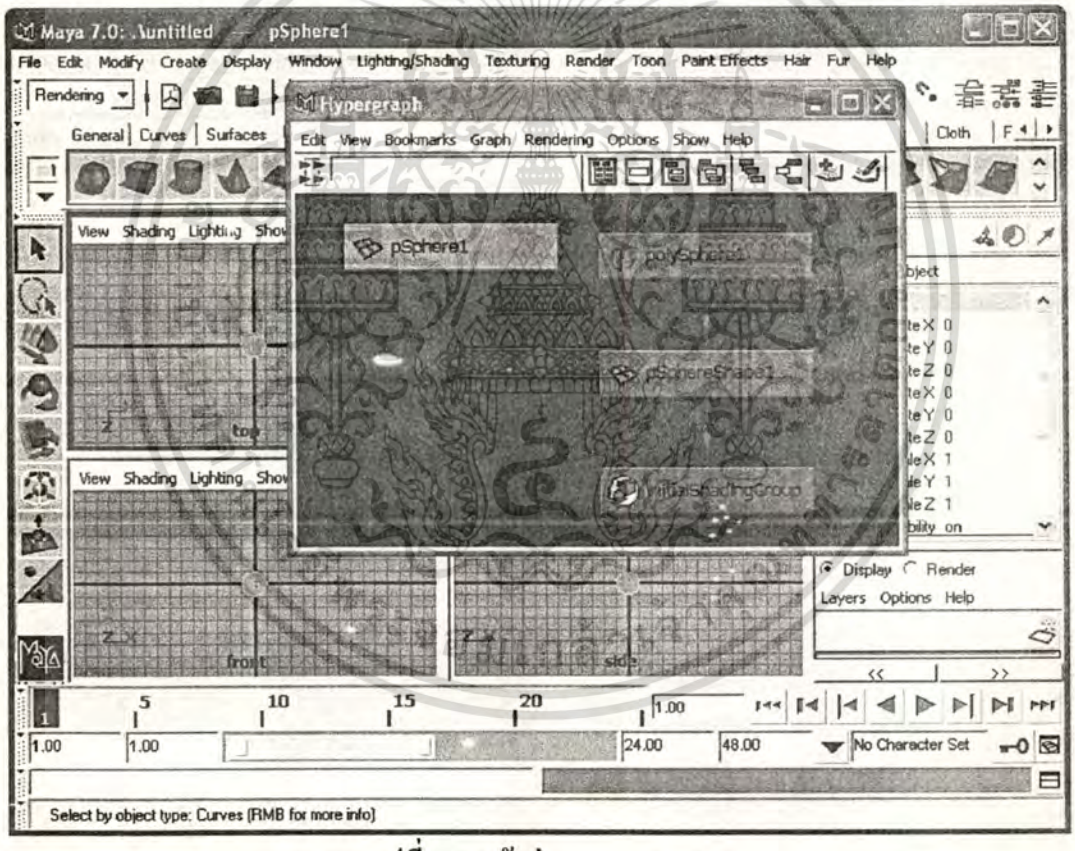
ทุกๆ สิ่งในมายาจะถูกแทนด้วยโหนด (Node) แล้วโหนดจะมีแอตทริบิวต์ (Attribute) ซึ่งสามารถถูกนำไปเชื่อมต่อ (Connect) เข้ากับแอตทริบิวต์ของโหนดอื่นได้ สถาปัตยกรรมแบบโหนดนี้จะยอมให้การเชื่อมต่อ (Connection) ถูกทำขึ้นได้ระหว่างทุกสิ่งในมายา แอตทริบิวต์ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

โหนดจะเป็นตัวกำหนดว่าสิ่งไหนเป็นเชป (Shape) ตำแหน่ง (Position) ประวัติการสร้าง (Construction history) และการทำเฉด (Shading) ของวัตถุ (Object) และด้วยสถาปัตยกรรมนี้ ผู้ใช้สามารถสร้างความสัมพันธ์ระหว่างวัตถุ (inter-object dependencies) ความสัมพันธ์ของกลุ่มเฉด (shading group dependencies) และทำการเชื่อมต่อ (Connect) เข้ากับโหนดของผู้ใช้เองได้

กราฟดีเพนเดนซี (Dependency Graph) เป็นแหล่งสะสมของโหนดต่างๆซึ่งถูกเชื่อมต่อเข้าด้วยกันโดยใช้แอตทริบิวต์ (Attribute) และการเชื่อมต่อเหล่านี้ยอมให้ข้อมูลมีการเคลื่อนที่จากโหนดหนึ่งไปสู่อีกโหนดหนึ่ง และสามารถถูกมองได้ในรูปแบบของไดอะแกรม โดยใช้หน้าต่างไฮเพอกราฟ (Hypergraph Window) ดังรูปที่ 3.3



รูปที่ 3.3 หน้าต่าง Hypergraph

ดีเพนเดนซีของโหนด (Node Dependencies) เป็นเส้นเชื่อมต่อระหว่างโหนดเพื่อแสดงทิศทางการเชื่อมต่อ เส้นเชื่อมต่อนี้มีจุดเริ่มต้นที่โหนดซึ่งส่งข้อมูลออก และเส้นนี้ชี้ไปที่โหนดซึ่งรับข้อมูลเข้า

ดีเพนเดนซี (Dependencies) เป็นความสัมพันธ์ (Relationship) ที่ถูกสร้างระหว่างโหนดซึ่งถูกเชื่อมต่อกัน ในหน้าต่างไฮเพอกราฟ (Hypergraph Window) ผู้ใช้สามารถดูดีเพนเดนซีของ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด (Node dependency) ที่ถูกเลือกไว้ โดยใช้ไฮเพอกราฟ (Hypergraph) ซึ่งจะแสดงการเชื่อมต่อระหว่างโหนดโดยจะมีลูกศรแสดงทิศทางของการดีเพนเดนซี (Dependency) ของโหนดเหล่านั้นที่มีต่อโหนดอื่น เช่น โพลีกอนรูปทรงกลม (Polygon Sphere) มีกราฟดีเพนเดนซีดังรูปที่ 3.4

รูปที่ 3.4 กราฟดีเพนเดนซีของโพลีกอนรูปทรงกลม

จากรูปที่ 3.4 โหนดชื่อ polySphere1 เป็นโหนดแบบโพลีสเฟียร์ (polySphere) ส่งข้อมูลไปที่โหนด pSphereShape1 ซึ่งเป็นโหนดแบบเมช (mesh) จากนั้นโหนด pSphereShape1 ส่งข้อมูลไปที่โหนด initialShadingGroup ซึ่งเป็นโหนดแบบเฉดดิงเอนยิน (shadingEngine) ซึ่งทั้ง 3 โหนดนี้ประกอบเป็นโหนด pSphere1 ซึ่งเป็นโหนดแบบ ทรานสฟอร์ม (transform)

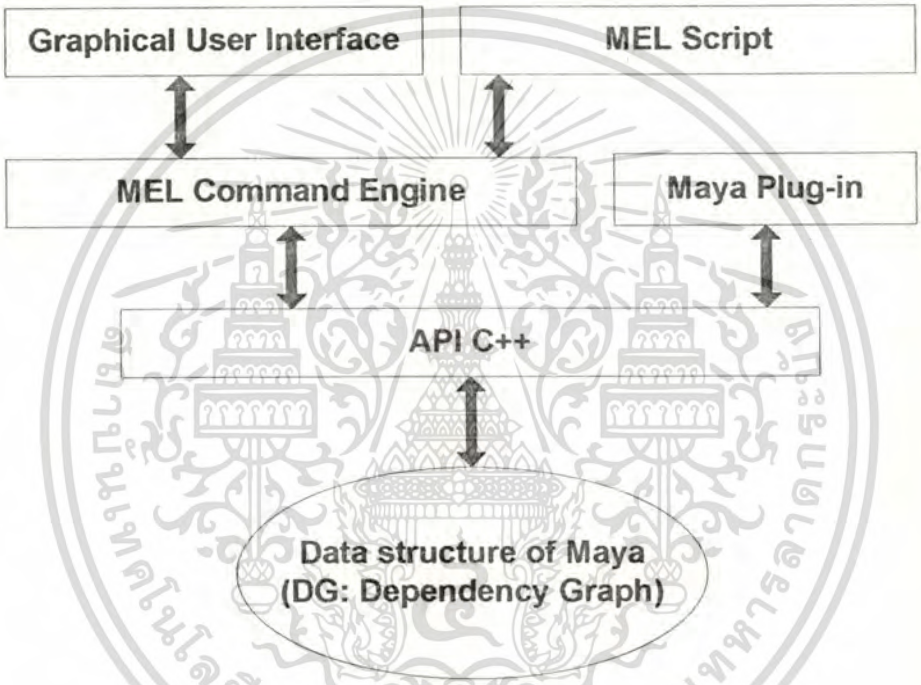
3.3 แนวทางในการพัฒนาโปรแกรมเสริม

มี 2 แนวทางในการพัฒนาโปรแกรมเสริมสำหรับมายา (Maya) คือ MEL และ Maya API แนวทางที่ใช้ MEL (Maya Embedded Language) คือการพัฒนาจะทำเป็นไฟล์ที่รวบรวมคำสั่งการทำงานหลายๆ คำสั่งเข้าด้วยกันเป็นไฟล์นามสกุล .mel ส่วนแนวทางการพัฒนาด้วย Maya API คือการพัฒนาจะใช้ Maya API ซึ่งเป็นไลบรารี (Library) ซึ่งโปรแกรมมายาได้จัดเตรียมไว้ให้ การใช้งานจะต้องมีการคอมไพล์ (Compile) ด้วยคอมไพลเลอร์ (Compiler) ภาษา C++ ซึ่งในโครงการนี้ใช้ Microsoft Studio.NET 2003 Visual C++ ผลที่ได้จากการคอมไพล์จะถูกลิงค์ (Linked) จนในที่สุดจะได้เป็นไฟล์นามสกุล .mll ทั้งสองแนวทางนี้ล้วนมีความสำคัญและเหมาะสมกับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างกัน ดังนั้นการพัฒนาโปรแกรมเสริมในรายงานนี้จึงใช้ทั้งสองแนวทาง โดยในส่วนของการติดต่อผู้ใช้ (User Interface) จะพัฒนาด้วย MEL และส่วนของการประมวลผล (Processing) จะพัฒนาด้วย Maya API ซึ่งสามารถทำงาน (Execute) เร็วกว่า MEL ถึง 10 เท่า

ลักษณะสถาปัตยกรรมของโปรแกรมมายาที่ทำงานร่วมกับโปรแกรมเสริมทั้งสองแบบแสดงได้ดังรูปที่ 3.5 โดย MEL Script จะทำงานผ่าน MEL Command Engine แต่ Maya Plug-in จะทำงานผ่าน API C++ ดังนั้นการเข้าถึงข้อมูลของมายาด้วย Maya Plug-in จะทำได้เร็วกว่าแบบ MEL



รูปที่ 3.5 ระบบมายาและโปรแกรมเสริมมายา

3.3.1 การพัฒนาโปรแกรมเสริมด้วย MEL

MEL เป็นภาษาสคริปต์ (Scripting Language) ซึ่งสืบทอดมาจากเชลล์สคริปต์ของระบบปฏิบัติการยูนิกซ์ (UNIX Shell Script) ดังนั้นคำสั่ง MEL ส่วนใหญ่จึงมีลักษณะเหมือนโปรแกรมยูทิลิตี (Utility Program) แบบคอมแมนด์ไลน์บนระบบปฏิบัติการยูนิกซ์ (UNIX Command-line) นั่นคือมีชื่อโปรแกรมแล้วก็ตามด้วยออปชัน (Option) ต่างๆใช้ในการปรับแต่งการทำงาน เช่น polySphere –r 5; หมายถึงสร้างโพลีกอนทรงกลมรัศมี 5 หน่วย เป็นต้น

MEL มีคำสั่งที่เกี่ยวกับการสร้าง GUI (Graphical User Interface) มากมายทำให้การเขียนโปรแกรมบนมายาแบบมี GUI สะดวกมาก และจากสถาปัตยกรรมของมายา แสดงว่าคำสั่งใดๆ ก็ตามที่ทำใน Graphical User Interface ของโปรแกรมมายานั้นจะไปเอ็กซิคิวต์ MEL Script อีกทอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่ง เช่น ในการสร้างโพลีกอน (Polygon) รูปทรงกลม (Sphere) เมื่อใช้คำสั่งจากเมนู (Menu) มายา ว่า Create > Polygon Primitives > Sphere จะมีการสร้าง MEL Script และเอ็กซิคิวต์โคดอัตโนมัติ ดังนี้

```
polySphere -r 1 -sx 20 -sy 20 -ax 0 1 0 -tx 1 -ch 1;
// Result: pSphere1 polySphere1 //
```

ผลที่ได้คือเป็นรูปโพลีกอนทรงกลมซึ่งมีรายละเอียดดังนี้

- -r 1 คือ รัศมี (Radius) 1 หน่วย
- -sx 20 คือ ค่าความละเอียดแนวแกน x (Subdivision Axis) เป็น 20
- -sy 20 คือ ค่าความละเอียดแนวแกน y (Subdivision Height) เป็น 20
- -ax 0 1 0 คือ แกน x = 0, y = 1, z = 0
- -tx 1 คือให้มีผิว (Texture)
- -ch 1 คือ เก็บประวัติการสร้าง

3.3.1.1 ค่าและตัวแปร (Values and Variables)

ค่าพื้นฐานแบ่งเป็น 4 ชนิดคือ

(1) เลขจำนวนเต็ม เช่น 5, -20, 0

(2) เลขทศนิยม เช่น 2.0, -6.3, 0.0

(3) เลขที่ไม่ใช่ฐานสิบ เช่น 0xA0

(4) สตริง (String) เช่น "MEL is fun!"

การกำหนดชนิดของค่าทำได้ 2 แบบคือแบบชี้ชัด (Explicit) และแบบบอกเป็นนัย (Implicit)

ตัวอย่างแบบชี้ชัด เช่น (float) 500 หมายถึง 500.0, (int) 7.5 หมายถึง 7

ตัวอย่างแบบบอกเป็นนัย เช่น "500" + 5 หมายถึง "5005", และ 5 + "500" หมายถึง "5500"

สำหรับตัวแปรนั้นจะต้องมีเครื่องหมายดอลลาร์ ($) นำหน้าเสมอ และจะต้องมีการประกาศชนิดก่อนใช้งาน เช่น

```
float $param;  int $counter;
string $name;  vector $position;
```

3.3.1.2 อาร์เรย์ (Arrays), เวคเตอร์ (vectors) และเมตริกซ์ (matrices)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาร์เรย์ (array) เป็นลิสต์ของค่าที่เรียงลำดับกัน และทุกค่าในอาร์เรย์จะต้องมีชนิดเดียวกัน

เวคเตอร์ (vector) เป็นเซตของค่าจำนวนจริง 3 ค่า ซึ่งมักอ้างถึง ค่า x, y, z ซึ่งเป็นโครงสร้างข้อมูลที่เหมาะกับคำสั่งในงาน 3 มิติเพราะเกี่ยวกับค่าของ x, y, z เช่น

```
vector $test = <<3.0,7.7,9.1>>;
print($test.x); // 3.0
print($test.y); // 7.7
print($test.z); // 9.1
```

เมตริกซ์ (Matrix) เป็นตาราง 2 มิติของค่าจำนวนจริง โดยอ้างเป็นแถว (Row) และสดมภ์ (Column) เช่น

```
matrix $a[3][4] = <<2.5,4.5,3.25,8.05;
                    1.12,1.3,9.5,5.2;
                    7.23,6.06,2.34,4.67>>
print($a[2][2]);          // 2.34
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.3 บทบาทร่วมกับ Maya API

ผู้ใช้สามารถเพิ่มคำสั่งหรือโหนดใหม่ๆ ให้มายาโดยใช้ Maya API และจากนั้นก็สร้างสคริปต์ MEL ให้สร้าง User Interface สำหรับคำสั่งหรือโหนดใหม่นั้นๆ เช่น ผู้ใช้สร้างโหนดชนิดใหม่ด้วย Maya API จากนั้นเขียนคำสั่งใหม่ซึ่งรู้วิธีที่จะสร้างอินสแตนท์ (Instance) ของโหนดนั้นแล้วโหลดคำสั่งนั้นลงในซีน (Scene) จากนั้นผู้ใช้สามารถเขียนสคริปต์ MEL ซึ่งแทรกคำสั่งนั้นไปบนเมนู (Menu) ของมายาเพื่อให้สามารถเลือกคำสั่งนั้นได้สะดวก

3.3.2 การพัฒนาโปรแกรมเสริมด้วย Maya API

Maya API เป็น C++ API ที่มีความสามารถในการเข้าถึง (Access) ข้อมูลทุกส่วนในโปรแกรมมายาได้ โดยมีลักษณะเป็นไลบรารีต่างๆดังนี้

- OpenMaya ประกอบด้วยคลาส (Classes) สำหรับการนิยามโหนด (Nodes) และคำสั่ง (Commands) และคลาสสำหรับการสร้างโปรแกรมเสริมจากโหนดและคำสั่งเหล่านี้
- OpenMayaUI ประกอบด้วยคลาสที่จำเป็นสำหรับการสร้างส่วนติดต่อกับผู้ใช้ขึ้นมาใหม่ เช่น Manipulators, Contexts , และ Locators
- OpenMayaAnim ประกอบด้วยคลาสสำหรับแอนิเมชัน (Animation) รวมถึงดีฟอร์เมอร์ (Deformers) และ Inverse Kinematics
- OpenMayaFX ประกอบด้วยคลาสสำหรับไดนามิกส์ (Dynamics)
- OpenMayaRender ประกอบด้วยคลาสสำหรับการทำฟังก์ชันการแสดงผล (Rendering Functions)

ในกรณีที่พัฒนาบนระบบปฏิบัติการ Windows คอมไพเลอร์สำหรับ Maya API คือ MS Visual Studio .NET 2003 Visual C++ สำหรับไลบรารี Maya API เหล่านี้จะถูกเก็บอยู่ในโฟลเดอร์ (Folder) ชื่อ C:\Program Files\Alias\Maya7.0\lib โดยไฟล์ไลบรารีเหล่านี้จะมีนามสกุล .lib ซึ่งจะต้องนำมาลิงค์เข้ากับโปรแกรมเพื่อให้ได้ไฟล์นามสกุล .mll และเพื่อความสะดวกในการสร้างโปรแกรมเสริม โปรแกรมมายาจึงได้เตรียม Maya Plug-in Wizard เอาไว้เพื่อสร้างไฟล์โปรเจ็ค (Project File) ของโปรแกรมเสริม ส่วนไฟล์ Wizard นี้จะอยู่ในโฟลเดอร์ C:\Program Files\Alias\Maya7.0\devkit\pluginwizard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.1 การปฏิสัมพันธ์กับโปรแกรมมายาด้วย Maya API

ใน Maya API มี Objects ซึ่งใช้ในการปฏิสัมพันธ์กับโปรแกรมมายา คือ Objects Function Sets Wrappers และ Proxies

โดยทั่วไปแล้ว Object จะประกอบด้วย 2 ส่วน คือ Attributes และ Function Sets แต่ใน Maya API นั้น Object จะมีแต่ Attributes ไม่มี Function Set โดยส่วน Function Set จะแยกไปเป็นอีกคลาสหนึ่ง เมื่อรวมกันระหว่าง Object และ Function Set ใน Maya API จึงคล้ายกับการมีสิ่งหนึ่งห่อหุ้มอีกสิ่งหนึ่งไว้ ทั้ง Objects และ Function Set นี้มีความเป็นเจ้าของ (Ownership) ต่างกัน กล่าวคือ Objects มีระบบโปรแกรมมายาเป็นเจ้าของเสมอ ส่วน Function Set เราจะเป็นเจ้าของเสมอ ซึ่งความเป็นเจ้าของนั้นเป็นสิ่งสำคัญของ Object ใน API เพราะความเป็นเจ้าของจะป้องกันความผิดพลาดจากการเขียนโปรแกรมที่ไม่ถูกต้อง ซึ่งอาจจะลบบางสิ่งที่ระบบต้องการ หรืออาจจะใช้บางสิ่งที่ระบบเพิ่งลบไป

ทุก Object ภายในโปรแกรมมายาเช่น Curves, Surfaces, DAG nodes, Dependency Graph Nodes, Lights, Shaders, และ Textures เป็นต้น ล้วนสามารถเข้าถึงได้จาก Handle Object ชื่อ MObject ซึ่งมี Method ที่ใช้ในการกำหนดชนิดของ Object ที่ต้องการอ้างถึง และการเรียกฟังก์ชัน Destructor ของ MObject จะลบเฉพาะ Object ตัวเอง แต่จะไม่ลบ Object ของโปรแกรมมายาที่ถูกอ้างถึง ซึ่งเป็นคุณสมบัติความเป็นเจ้าของ (Ownership)

สำหรับ Wrappers เป็นอาร์เรย์ของ Objects (Array of Objects) เช่น อาร์เรย์ของ Object ชื่อ MPoint มีคลาส MPointArray เป็น Wrapper

ส่วน Proxies เป็น Objects ซึ่ง Maya API ใช้เพื่อสร้างชนิดของ Maya Object ใหม่ ดังนั้น Proxies จึงเป็น Objects ที่เราสร้างขึ้น แต่ระบบโปรแกรมมายาได้ความเป็นเจ้าของ

สำหรับการตั้งชื่อคลาสของ Maya API จะใช้ Prefix ซึ่งแสดงถึง Objects ต่างๆ คือ

- MFn แสดงว่าเป็น Function Set ซึ่งถูกใช้กับ MObjects
- MIt แสดงว่าเป็น Iterators ซึ่งถูกใช้กับ MObjects
- MPx แสดงว่าเป็น Proxies โดย เราสามารถสร้าง Maya Object ใหม่ โดยการรับจาก (Derive) คลาสที่มีอยู่เดิม
- M แสดงว่าเป็น Wrappers

3.3.2.2 ขั้นตอนการสร้างโปรแกรมเสริมด้วย Maya API

ในการเขียนโปรแกรมเสริมด้วย Maya 7.0 API นั้น มีสิ่งสำคัญที่จำเป็นต้องทราบคือ การลงทะเบียน (Register) กับมายา ความสำคัญของการลงทะเบียน (Register) คือทำให้มายารู้จัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเสริมและสามารถเรียกใช้โปรแกรมเสริมได้ ซึ่งการลงทะเบียนนี้สามารถทำได้โดยใช้มาโคร (Macro) DeclareSimpleCommand() ซึ่งถูกประกาศในไฟล์ Msimple.h หรือใช้เมธอด (Method) initializePlugin() ในคลาส MFnPlugin ซึ่งเป็นคลาสที่ถูกประกาศในไฟล์ MfnPlugin.h

3.3.2.3 ตัวอย่างคลาสของ Maya API

คลาสที่สำคัญของ Maya API เช่น MFnPlugin, MGlobal, MObject, MStatus ซึ่งผู้ใช้สามารถเข้าดูรายละเอียดได้ในโฟลเดอร์ (Folder) C:\Program Files\Alias\Maya7.0\include\maya

3.4 โปรแกรมเสริมของโครงการ

หน้าที่ของโปรแกรมเสริมของโครงการคือช่วยให้การสร้างภาพเคลื่อนไหวของปากในโปรแกรมมายาทำได้รวดเร็วขึ้น ระบบโปรแกรมเสริมนี้ประกอบด้วย 1) โมดูลหลัก 2) โมดูลย่อยการติดต่อกับผู้ใช้ 3) โมดูลย่อยการแปลงข้อความเป็นไฟล์หน่วยเสียงและไฟล์เสียง 4) โมดูลย่อยการสร้างรูปปาก 5) โมดูลย่อยการสร้างภาพเคลื่อนไหว มีเพียงโมดูลย่อยการติดต่อกับผู้ใช้เท่านั้นที่ถูกพัฒนาขึ้นด้วย MEL script ส่วนที่เหลืออื่นๆจะถูกพัฒนาขึ้นโดยใช้ Maya API

โมดูลหลักถูกออกแบบให้ใช้ Maya API โดยใช้ชื่อ Class เป็น want2talkthai เพื่อให้การประมวลผลรวดเร็วและการควบคุมระบบย่อยต่างๆ ทำได้ดี โมดูลย่อยการติดต่อกับผู้ใช้ถูกออกแบบให้เป็น MEL script เพราะพัฒนาง่ายและไม่ต้องใช้ความเร็วในการประมวลผลมาก โมดูลย่อยการแปลงข้อความเป็นไฟล์หน่วยเสียงและไฟล์เสียง ใช้ Maya API โดยใช้ชื่อ Class เป็น CVajaTTS โมดูลย่อยการสร้างรูปปาก ใช้ Maya API ใช้ชื่อ Class เป็น CLipShape ส่วนโมดูลย่อยการสร้างภาพการเคลื่อนไหวใช้ Maya API ชื่อ Class เป็น CLipAnimation

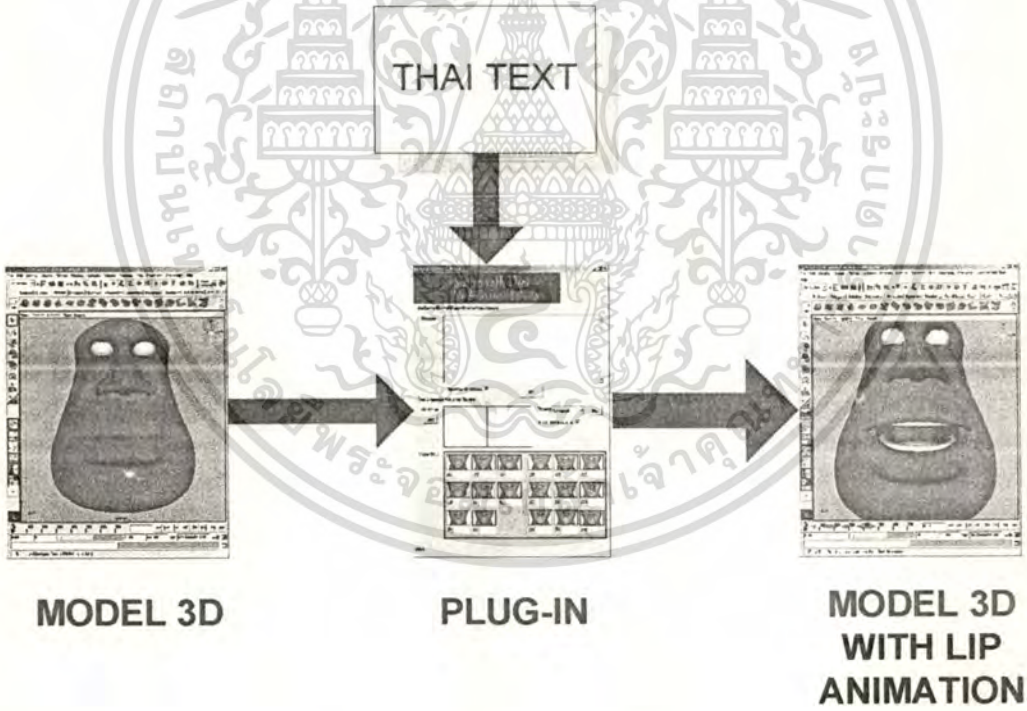
3.5 แนวคิดระบบโปรแกรมเสริมของโครงการ

แนวคิดในการออกแบบระบบโปรแกรมเสริมเปรียบเหมือนคนอ่านหนังสือ เมื่อมีข้อความให้อ่าน สิ่งแรกที่ทำให้เราสามารถอ่านได้คือต้องรู้จักตัวอักษร (Consonants) ต้องรู้วิธีการขยับปากให้ได้หน่วยเสียง (Phonemes) ของแต่ละตัวอักษร และต้องรู้วิธีสะกดหน่วยเสียงออกมาเป็นเสียงพยางค์ เมื่อเสียงพยางค์ก็เกิดจากการผสมกันของหลายหน่วยเสียง ดังนั้นรูปปากของเสียงพยางค์ก็เกิดจากการผสมกันของรูปปากหน่วยเสียงที่เกี่ยวข้องต่างๆ เหล่านั้นด้วย แต่การออกแบบระบบโปรแกรมเสริมนี้ไม่ได้คำนึงถึงเรื่องระดับการพูดเสียงดัง เสียงค่อย ดังนั้นการสร้างภาพเคลื่อนไหวของปากในการพูดของระบบจึงมีระดับเสียงดังเดียวคือระดับเสียงปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแนวคิดข้างต้นจึงได้นำมาออกแบบโมดูลหลัก เป็น Class คือ want2talkthai ซึ่งจะทำงานเมื่อมีการเรียกใช้โปรแกรมเสริมจากภายในโปรแกรมมายา want2talkthai เป็น Class ที่จะเป็นศูนย์กลางการควบคุมการทำงานของระบบโปรแกรมเสริมทั้งหมด ให้ทำงานตามอัลกอริธึม (Algorithm) ที่ได้ออกแบบไว้

โปรแกรมเสริมนี้ทำให้กระบวนการสร้างภาพเคลื่อนไหวของปากให้กับโมเดล 3 มิติ ทำได้รวดเร็วขึ้น โดยเริ่มจากผู้ใช้พิมพ์ข้อความเข้าไปแล้วโปรแกรมอ่านข้อความและแปลงข้อความเป็นหน่วยเสียงและไฟล์เสียงจากนั้นจึงจับคู่หน่วยเสียงเข้ากับรูปปาก และผสมรูปปากเป็นเสียงพยางค์จากนั้นจึงทำการขยับรูปปากให้เป็นการพูดเป็นพยางค์ๆตามข้อความที่ผู้ใช้ป้อนเข้าไป โดยใช้วิธีกำหนด Keyframe รูปปากในแต่ละช่วงเวลา จากนั้นจึง Render โมเดลให้เป็นไฟล์ .avi ก็จะได้ภาพการเคลื่อนไหวของปากของโมเดลนั้น ดังรูปที่ 3.6



รูปที่ 3.6 ลักษณะการทำงานของโปรแกรมเสริม

3.6 โครงสร้างข้อมูลและอัลกอริธึมโมดูลหลัก

ในการควบคุมการสร้างภาพเคลื่อนไหวของปากนั้น เป็นหน้าที่ของโมดูลหลักซึ่งควบคุมและสนับสนุนการทำงานของโมดูลย่อยอื่นๆ ตั้งแต่การเริ่มพิมพ์ข้อความจนถึงขั้นตอนสุดท้ายที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพการเคลื่อนไหวของปาก ดังนั้นจึงต้องพัฒนาโครงสร้างข้อมูลและอัลกอริธึมของโมดูลหลักขึ้นมาใหม่ให้สามารถควบคุมและสนับสนุนการทำงานของโมดูลย่อยอื่นๆ ได้

3.6.1 โครงสร้างข้อมูลโมดูลหลัก

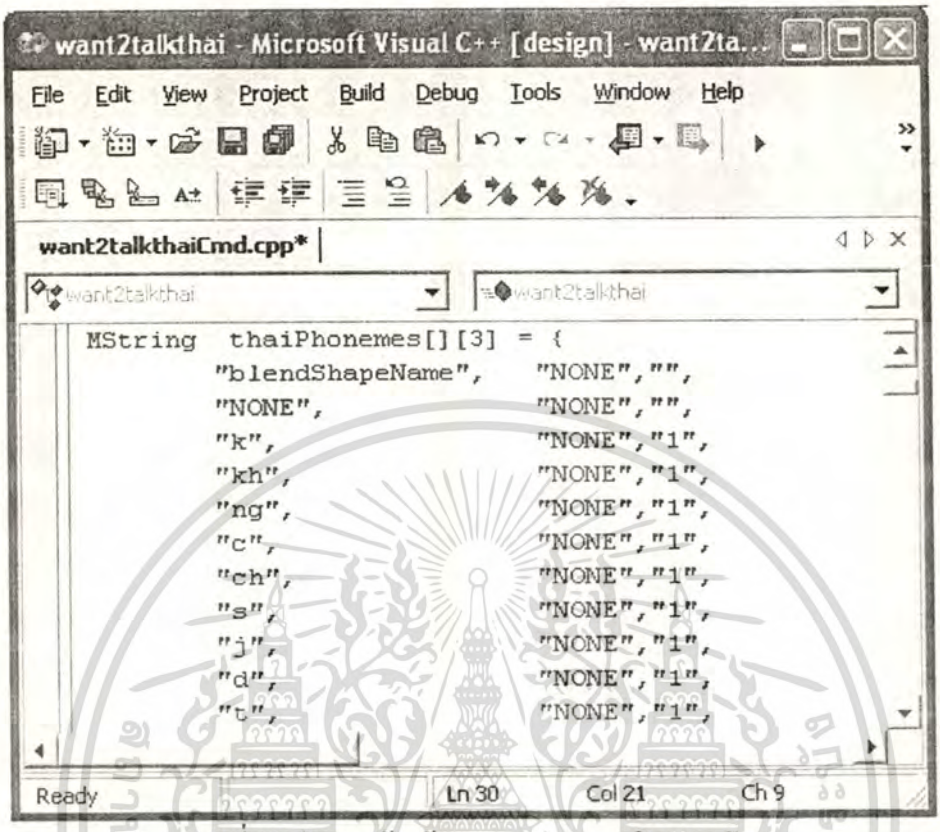
การออกแบบโครงสร้างข้อมูลของโมดูลหลักมีรูปแบบเป็น Array 2 มิติของตัวแปรชนิด MString โครงสร้างข้อมูลนี้มีหน้าที่เก็บค่าการจับคู่หน่วยเสียงกับรูปปากของโมเดลและค่าการจัดแบ่งกลุ่มของหน่วยเสียงที่มีรูปปากเดียวกัน การประกาศโครงสร้างข้อมูลเป็นดังนี้

```
MString thaiPhonemes[phoneme_No][3];
```

จากคำสั่ง เป็นการประกาศตัวตัวแปร thaiPhonemes เป็น Array 2 มิติชนิด MString โดยมิติแรกเก็บค่าแต่ละหน่วยเสียง มิติที่สองเก็บ 3 ค่า คือ ชื่อหน่วยเสียง ชื่อรูปปากของโมเดล และค่าการจัดแบ่งกลุ่มของหน่วยเสียงที่มีรูปปากเดียวกัน ตามลำดับ สำหรับข้อมูลเริ่มต้นของระบบโปรแกรมหลักจะตั้งค่าหน่วยเสียงของภาษาไทยจำนวน 60 หน่วยเสียง ประกอบด้วยเสียงพยัญชนะ 33 หน่วยเสียง และสระ 27 หน่วยเสียง เรียงตามลำดับที่แน่นอน ซึ่งลำดับนี้มีความสำคัญในการทำให้รู้ว่าจะเลือกโมเดลรูปปากใดสำหรับหน่วยเสียงที่ต้องการในขั้นตอนการรับค่าจากผู้ใช้

บางส่วนของโครงสร้างข้อมูลนี้แสดงในรูปที่ 3.7 คอลัมน์ (Column) แรกเป็นชื่อหน่วยเสียง คอลัมน์ที่สองเป็นชื่อรูปปาก ซึ่งถ้ายังไม่มีการจับคู่กับหน่วยเสียงจะมีค่าเป็น "NONE" ส่วนคอลัมน์ที่สามเป็นค่าการจัดแบ่งกลุ่มของหน่วยเสียงที่มีรูปปากเดียวกัน หมายเลขนี้มีประโยชน์คือทำให้รู้ว่าจะสามารถใช้รูปปากใดแทนซึ่งกันและกันได้ ทำให้เพิ่มความรวดเร็วในการประมวลผลและประหยัดทรัพยากรของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

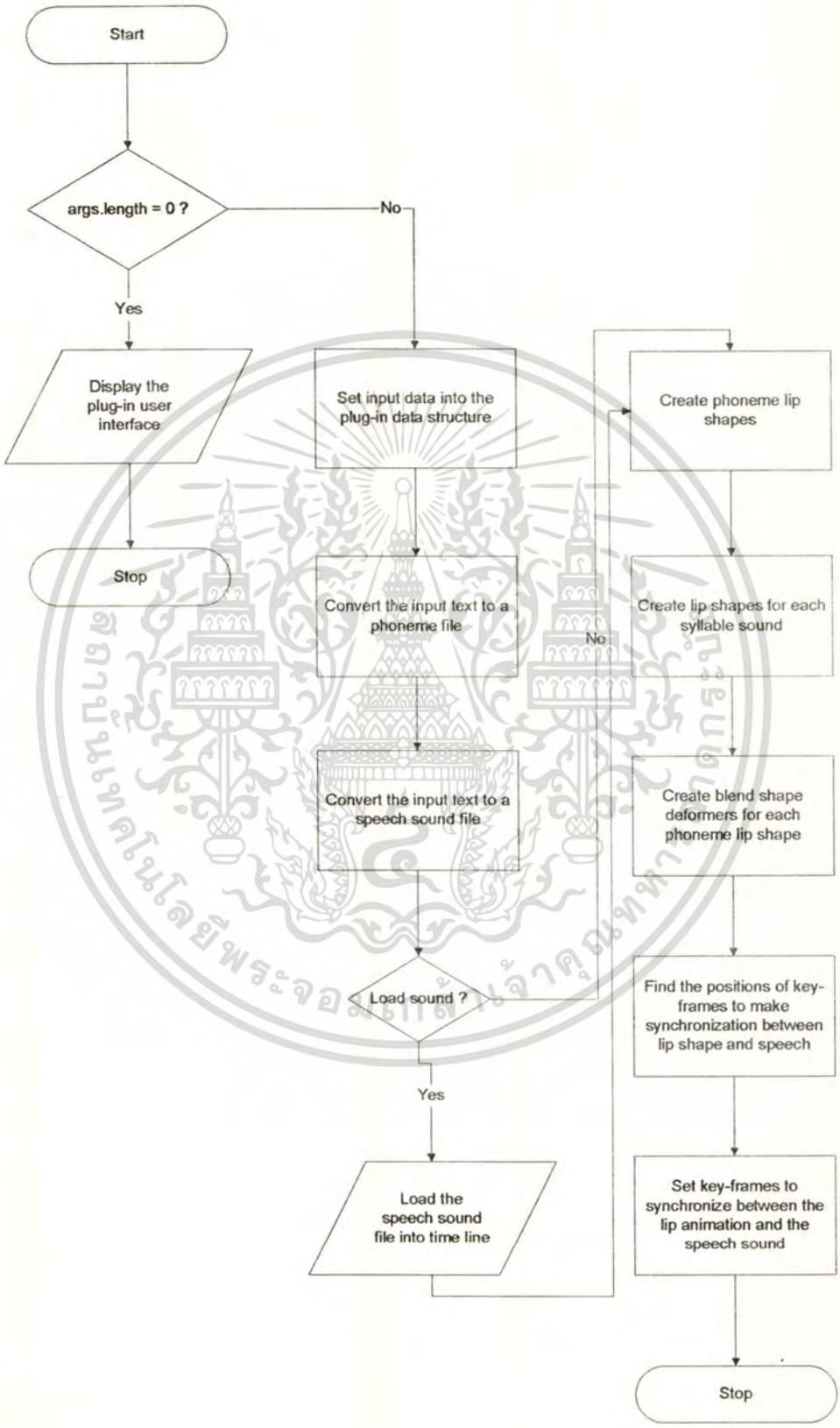


รูปที่ 3.7 โครงสร้างข้อมูลบางส่วนของโมดูลหลัก

3.6.2 อัลกอริธึมโมดูลหลัก

อัลกอริธึมของโมดูลหลักสามารถเขียนเป็น Flow chart ได้ตามหน้าที่ ดังรูปที่ 3.8 เริ่มจากเมื่อผู้ใช้เรียกใช้โปรแกรม 1) โมดูลหลักจะตรวจเช็คจำนวน Argument ที่ส่งให้ว่ามีค่าเป็น 0 หรือไม่ ถ้าเป็น 0 ให้แสดงส่วนติดต่อกับผู้ใช้เพื่อให้ผู้ใช้ใส่ข้อความและข้อมูลรูปปากของหน่วยเสียงแล้วจบการทำงาน แต่ถ้าไม่เท่ากับ 0 ทำข้อ 2 ต่อไป 2) กำหนดค่าข้อมูลรูปปากของหน่วยเสียงเข้ากับโครงสร้างข้อมูล thaiPhonemes 3) แปลงข้อความที่ผู้ใช้ป้อนเข้ามาให้เป็นไฟล์หน่วยเสียง 4) แปลงข้อความที่ผู้ใช้ป้อนเข้ามาให้เป็นไฟล์เสียงพูด 5) ตรวจสอบว่าต้องการโหลดไฟล์เสียงพูดที่สังเคราะห์ขึ้นนี้หรือไม่ ถ้าต้องการโหลด ให้ทำข้อ 6 ถ้าไม่ต้องการโหลดให้ทำข้อ 7 6) โหลดเสียงพูดที่สังเคราะห์ขึ้นใส่ Time line 7) สร้างรูปปากสำหรับแต่ละหน่วยเสียง 8) สร้างรูปปากสำหรับแต่ละพยางค์ 9) สร้าง Blend shape deformer สำหรับแต่ละพยางค์ 10) หาตำแหน่ง Keyframe เพื่อให้ภาพเคลื่อนไหวที่ได้ตรงกับเสียงพูด 11) กำหนดตำแหน่ง Keyframe ตามข้อ 10 แล้วจึงสิ้นสุดการทำงานของโมดูลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Flow chart อัลกอริธึมโมดูลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2.1 อัลกอริธึมในการแสดงส่วนติดต่อกับผู้ใช้

คำสั่งในการเรียกใช้โปรแกรมเสริมจากเมนูของโปรแกรมมายาจะมี 2 ลักษณะคือ ลักษณะไม่มีส่วนของ Arguments ซึ่งเป็น Option สำหรับคำสั่งนั้น กับลักษณะที่มี Argument ทั้งสองลักษณะมีประโยชน์ในการเรียกใช้ฟังก์ชันที่ต่างกันจากโปรแกรมเดียวกัน และเพื่อให้มีการเรียกใช้ฟังก์ชันแสดงส่วนติดต่อกับผู้ใช้ การเรียกคำสั่งจะต้องกำหนดว่าถ้าจำนวน Argument ไม่มากกว่า 0 ให้เรียกใช้คำสั่ง DisplayTheUI ซึ่งเป็นคำสั่งสำหรับสร้างและแสดงส่วนติดต่อกับผู้ใช้ อัลกอริธึมนี้เป็นดังรูปที่ 3.9

```
if(args.length()>0)
{
      …
}
else
{
      DisplayTheUI();
}
```

รูปที่ 3.9 อัลกอริธึมในการแสดงส่วนติดต่อกับผู้ใช้

3.6.2.2 อัลกอริธึมในการกำหนดค่าข้อมูลรูปปากของหน่วยเสียงให้กับโครงสร้างข้อมูล

ปัญหาในการส่งค่าระหว่างโมดูลย่อยส่วนติดต่อกับผู้ใช้และโมดูลหลักในโปรแกรมมายา คือค่าตัวแปรของ MEL script ซึ่งเป็น Process ส่วนติดต่อกับผู้ใช้ไม่สามารถนำไปให้กับ Process โมดูลหลักโดยตรงระหว่าง Processes ได้ การส่งค่าจากส่วนติดต่อกับผู้ใช้ไปยังโมดูลหลักจึงต้องทำผ่านไฟล์เฉพาะชื่อ "w2tt.txt" ซึ่งเป็นไฟล์ที่ถูกสร้างเอาไว้เก็บข้อมูลคำสั่งของโมดูลย่อยส่วนติดต่อกับผู้ใช้ ข้อมูลภายในไฟล์เป็นรหัสควบคุมซึ่งมีความหมายดังตารางที่ 3.1

ตารางที่ 3.1 รหัสควบคุมภายในไฟล์ w2tt.txt

รหัส	ความหมาย
1. -s	โหลดไฟล์เสียงพูดที่ได้จากการสังเคราะห์ใส่ใน Time line
2. –v	ส่วนที่ตามหลังเป็นข้อมูลรูปปากของหน่วยเสียงที่จับคู่ตรงกับรูปปากที่มีในโมเดลของผู้ใช้
3. -t	ส่วนที่ตามหลังเป็นข้อความที่ผู้ใช้พิมพ์เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเมื่อผู้ใช้ต้องการให้ขยับปากตามข้อความ "คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง" พร้อมทั้งให้มีเสียงพูด จะได้ไฟล์ข้อมูล w2tt.txt ซึ่งถูกสร้างขึ้นโดยส่วนติดต่อกับผู้ใช้ ดังรูปที่ 3.10

รูปที่ 3.10 ตัวอย่างข้อมูลไฟล์ w2tt.txt

เมื่อได้ไฟล์ w2tt.txt แล้ว โมดูลจะถูกเรียกขึ้นมาอีกครั้งเพื่อนำข้อมูลจากไฟล์นี้ไปกำหนดให้กับโครงสร้างข้อมูล thaiPhonemes ซึ่งอัลกอริธึมในการกำหนดค่าคือ 1) อ่านตัวอักษรจากไฟล์ w2tt.txt 2) ถ้าพบตัวอักษรสองตัวซึ่งเป็นรหัสควบคุมการโหลดเสียง (-s) ให้เซตค่าตัวแปร importSound = true จากนั้นทำข้อ 1 ใหม่เป็นการอ่านตัวอักษรถัดไป 3) ถ้าพบตัวอักษรสองตัวซึ่งเป็นรหัสควบคุมรูปปาก (-v) ให้นำค่าถัดจากรหัสควบคุมทั้งหมดใส่ในโครงสร้างข้อมูล thaiPhonemes[i][1] ซึ่งเป็นการจับคู่หน่วยเสียงกับรูปปากที่ผู้ใช้กำหนดให้ เมื่อพบรหัสคำสั่งควบคุมอื่นให้หยุดแล้วทำข้อ 1 ใหม่ 4) ถ้าพบตัวอักษรสองตัวซึ่งเป็นรหัสควบคุมข้อความนำเข้า (-t) ให้นำค่าถัดจากรหัสควบคุมนั้นไปเก็บในบัฟเฟอร์ข้อความสำหรับใช้ในขั้นตอนอื่นต่อไป ข้อดีของอัลกอริธึมส่วนนี้คือผู้ใช้สามารถใส่ข้อความได้โดยไม่จำกัดตัวอักษร ซึ่งสามารถเขียนได้ดังรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
read_file_to_buffer("w2tt.txt", cmdBuffer);
istext = isphoneme=false;
j = ti = 0;
b = &cmdBuffer[0];

for (; *b!=NULL;b++ )
{
  if ( *b == '-' && *(b+1)=='s' && *(b+2)==' ')
  {
            importSound = true;
            b+=2;
  }
  else if ( *b == '-' && *(b+1)=='v' && *(b+2)==' ' )
  {
            isphoneme = true;
            b+=2;
  }
  else if ( *b == '-' && *(b+1)=='t' && *(b+2)==' ')
  {
            istext = true;
            isphoneme = false;
            b+=2;
  }
  else
  {
            if(isphoneme)
            {
                char  wd[256];
                for(int d=0; *b!=' ';d++,b++)
                  wd[d] =  *b;
                wd[d] = '\0';
                thaiPhonemes[j][1] = wd;
                j++;
            }
            else if(istext)
            {
                t[ti] = *b;
                ti++;
            }
            else {}
  }
}
t[ti] = '\0';
```

รูปที่ 3.11 อัลกอริธึมการกำหนดค่าโครงสร้างข้อมูล thaiPhonemes

3.6.2.3 อัลกอริธึมในการแปลงข้อความให้เป็นไฟล์หน่วยเสียงและไฟล์เสียงพูด

ในการแปลงข้อความซึ่งอยู่ในบัฟเฟอร์ให้กลายเป็นไฟล์หน่วยเสียงและไฟล์เสียงพูด เป็นการเรียกใช้ฟังก์ชันจากโมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง ซึ่งรายละเอียดจะกล่าวถึงในบทของโมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2.4 อัลกอริธึมในการโหลดไฟล์เสียงพูด

การตัดสินใจว่าจะโหลดไฟล์เสียงพูดที่ได้จากการสังเคราะห์เข้าไปใส่ใน Time line หรือไม่ จะขึ้นอยู่กับค่าตัวแปร importSound ซึ่งถูกกำหนดในขั้นตอนการกำหนดค่าข้อมูลรูปปากของหน่วยเสียงให้กับโครงสร้างข้อมูลว่ามีค่าเป็น true หรือไม่ ถ้าเป็น true ระบบโปรแกรมหลักจะโหลดไฟล์เสียงชื่อ "z.wav" ลงใน Time line แต่ปัญหาของขั้นตอนนี้คือ หากโหลดไฟล์เสียงทันทีโดยไม่รอให้ Threads ที่เกิดจากฟังก์ชันในการแปลงข้อความเป็นหน่วยเสียงและฟังก์ชันการแปลงข้อความเป็นไฟล์เสียงทำงานเสร็จสิ้นก่อนจะทำให้ระบบหยุดทำงาน (System crash) ดังนั้นก่อนการโหลดจึงต้องรอให้แน่ใจว่า Threads ดังกล่าวทำงานเสร็จเรียบร้อยแล้วจึงโหลดไฟล์เสียง อัลกอริธึมนี้แสดงดังรูป 3.12

```
if(importSound)
{
  int st;
  st = ConvertText.getstatus();
  while(st==VSTATUS_PLAYING)
  {
     st = ConvertText.getstatus();
  }
  executeCommand("want2talkthaiLoadAudioFile z.wav");
}
```

รูปที่ 3.12 อัลกอริธึมการโหลดไฟล์เสียงพูด

3.6.2.5 อัลกอริธึมในการสร้างรูปปากของหน่วยเสียง, เสียงพยางค์, และ การสร้าง Blend Shape Deformation

อัลกอริธึมทั้ง 3 นี้ไม่ได้ทำงานที่โมดูลหลัก แต่การสร้างรูปปากสำหรับแต่ละหน่วยเสียง, เสียงพยางค์, และการสร้าง Blend shape deformation จะเรียกใช้โมดูลย่อยการสร้างรูปปาก ซึ่งรายละเอียดจะได้กล่าวต่อไปในบทของโมดูลย่อยการสร้างรูปปาก

3.6.2.6 อัลกอริธึมในการหาตำแหน่งเวลาที่ใช้ในการกำหนดรูปปากและการกำหนดค่าควบคุมการเคลื่อนไหวของปาก

อัลกอริธึมทั้งสองนี้ไม่ได้ทำงานที่โมดูลหลัก การหาตำแหน่งเวลาที่ใช้ในการกำหนดรูปปากที่เหมาะสมและการกำหนดค่า Keyframe เพื่อควบคุมการเคลื่อนไหวของปาก จะเรียกใช้โมดูลย่อยการสร้างภาพเคลื่อนไหวของปาก ซึ่งจะกล่าวถึงต่อไปในบทโมดูลย่อยการสร้างภาพเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 สรุป

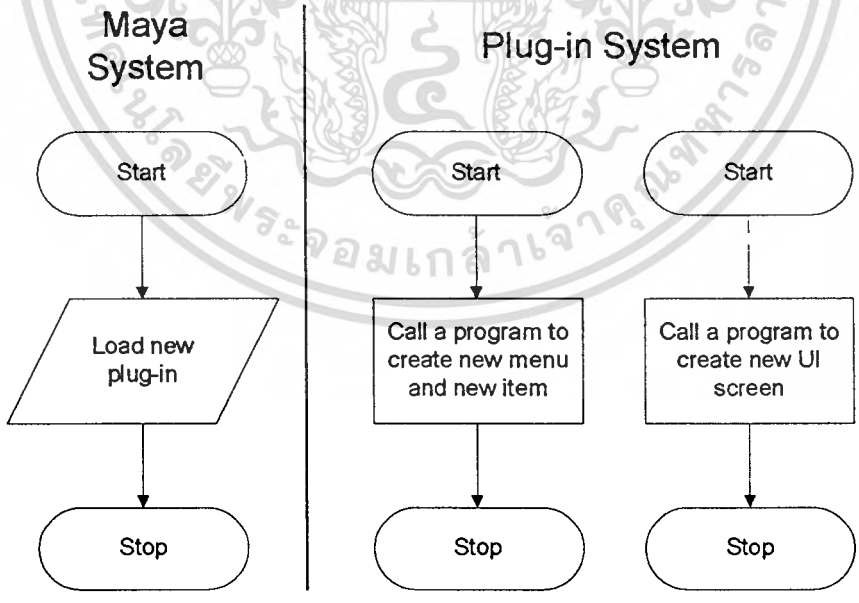
โปรแกรมมายาจัดเก็บข้อมูลด้านกราฟิกส์อยู่ในรูปของ Dependency Graph ระบบโปรแกรมเสริมสำหรับโปรแกรมมายามี 2 แบบคือ MEL script และ Maya API ส่วนระบบโปรแกรมเสริมที่ถูกพัฒนาขึ้นนี้ใช้ทั้งสองแบบในการสร้าง โดยมีส่วนโมดูลหลักเป็นส่วนควบคุมการทำงานและใช้ Maya API ขณะที่ส่วนติดต่อกับผู้ใช้และการโหลดไฟล์เสียงใช้ MEL script

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โมดูลย่อยส่วนติดต่อกับผู้ใช้

4.1 การสร้างโมดูลย่อยส่วนติดต่อกับผู้ใช้

ในภาพรวมแล้ว โมดูลย่อยส่วนติดต่อกับผู้ใช้ของโปรแกรมเสริมจะมีหน้าที่ในการรับข้อความนำเข้าและข้อมูลรูปปากสำหรับแต่ละหน่วยเสียงจากผู้ใช้ แล้วส่งให้กับโมดูลหลัก แต่การทำงานที่อยู่เบื้องหลังของโมดูลย่อยนี้ ไม่ใช่ทำได้โดยโปรแกรมเดียวและขั้นตอนเดียว หากแต่มีหลายมีหลายโปรแกรมและหลายขั้นตอนที่ทำงานร่วมกัน คือเมื่อเริ่มต้นโปรแกรมมายาโหลดโปรแกรมเสริมใหม่แล้วจบการทำงาน โมดูลหลักจะเรียกคำสั่งสร้างเมนูสำหรับโปรแกรมเสริม หลังจากนั้นจะมีเมนูใหม่เกิดขึ้นในโปรแกรมมายา เมื่อผู้ใช้เลือกเมนูนั้นจะเป็นการเรียกโปรแกรมหลักอีกครั้ง โดยครั้งนี้โมดูลหลักจะเรียกอีกโปรแกรมหนึ่งให้ทำการแสดงส่วนติดต่อกับผู้ใช้แล้วโมดูลหลักก็จบการทำงาน ดังรูปที่ 4.1



รูปที่ 4.1 การสร้างโมดูลย่อยส่วนติดต่อกับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โปรแกรมสร้างเมนูใหม่สำหรับโปรแกรมเสริม

โดยปกติการสร้างโปรแกรมเสริมขึ้นมาใหม่สำหรับโปรแกรมใดๆ จะมีวิธีในการสร้างเมนูขึ้นมาใหม่เพื่อให้สามารถเรียกใช้งานโปรแกรมเสริมนั้นได้ สำหรับในการสร้างเมนูเรียกใช้งานโปรแกรมเสริมมายาทำได้โดยใช้คำสั่งของ MEL script คือ menu ซึ่งเป็นคำสั่งในการสร้างเมนูใหม่และเพิ่มเข้าไปในเมนูโปรแกรมมายาและ menuItem ซึ่งเป็นคำสั่งในการสร้าง Menu items สำหรับเมนูใหม่ รูปที่ 4.2 แสดงโปรแกรมการสร้างเมนูใหม่ want2talkthaiMenu.mel สำหรับโปรแกรมเสริม

```
global proc want2talkthaiMenu()
{
  global string $gMainWindow;
  setParent $gMainWindow;
  menu -label "Want to Talk Thai" myNewMainMenu;

  setParent -menu myNewMainMenu;
  menuItem -label "Lip Animation Options" -command
      want2talkthai;
}
```

รูปที่ 4.2 โปรแกรมสร้างเมนูใหม่ของโปรแกรมเสริม

4.3 โปรแกรมสร้างหน้าจอรับค่าข้อมูลใหม่สำหรับโปรแกรมเสริม

โปรแกรมส่วนนี้ชื่อ want2talkthaiOptions.mel เป็นส่วนที่มีหน้าที่ในการรับข้อความนำเข้าและข้อมูลรูปปากสำหรับแต่ละหน่วยเสียงจากผู้ใช้จริงๆ โปรแกรมนี้จะส่งค่าข้อมูลที่ได้ให้กับโปรแกรมหลักโดยการเขียนลงไฟล์ข้อความชื่อ "w2tt.txt" สำหรับการส่งข้อมูลจากโปรแกรมแบบ MEL ไปให้โปรแกรมแบบ Maya API นั้นสามารถทำได้อีกวิธี คือโดยการเรียกใช้โปรแกรมแบบ Maya API แล้วตามด้วย Argument ข้อมูลที่ต้องการส่งให้กับโปรแกรม แต่เหตุผลที่การพัฒนาโครงการนี้ไม่ส่งข้อมูลด้วยวิธีนี้เพราะว่ามีข้อจำกัดเรื่องขนาดความยาวของข้อมูลในส่วนที่เป็น Argument ซึ่งถ้าเป็นการส่งในลักษณะไฟล์จะไม่จำกัดในเรื่องนี้

4.3.1 โพรโตคอล (Protocol) ในการส่งข้อมูลให้โมดูลหลักโดยการสร้างไฟล์

ไฟล์ที่จะใช้เป็นสื่อกลางในการส่งข้อมูลให้กับโมดูลหลักนี้ชื่อ w2tt.txt ซึ่งจะต้องถูกสร้างโดยใช้โพรโตคอลเดียวกับการอ่านไฟล์ของโมดูลหลัก เพื่อให้โปรแกรมติดต่อผู้ใช้และโปรแกรมเสริมนั้นเข้าใจข้อมูลได้ตรงกัน โพรโตคอลนี้มีรหัสควบคุมแสดงดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 รหัสควบคุมของโพรโตคอลการส่งข้อมูลเป็นไฟล์ w2tt.txt

รหัส	ความหมาย
1. –s	ให้มีการโหลดไฟล์เสียงพูดที่ได้จากการสังเคราะห์ใส่ใน Time line
2. –v	ส่วนที่ตามหลังเป็นข้อมูลรูปปากของหน่วยเสียงที่จับคู่ตรงกับรูปปากที่มีในโมเดลของผู้ใช้
3. –t	ส่วนที่ตามหลังเป็นข้อความที่ผู้ใช้พิมพ์เข้ามา

ข้อมูลที่ตามหลังรหัสควบคุม –v จะมีลำดับเป็นสิ่งสำคัญในการแสดงความหมายของการจับคู่ระหว่างหน่วยเสียงและรูปปากสำหรับหน่วยเสียงนั้น โดยมีหน่วยเสียงที่สามารถให้เลือกจับคู่ได้ดังตารางที่ 4.2

ตารางที่ 4.2 ความหมายลำดับที่ตามหลังรหัสควบคุม –v

ลำดับที่	ความหมาย
1	ชื่อ Blend shape ของโมเดล
2	ไม่มีการจับคู่รูปปาก (Unmapped)
3	ชื่อรูปปากหน่วยเสียง /ก/k/
4	ชื่อรูปปากหน่วยเสียง /ข,ฃ,ค,ฅ,ฆ/kh/
5	ชื่อรูปปากหน่วยเสียง /ง/ng/
6	ชื่อรูปปากหน่วยเสียง /จ/c/
7	ชื่อรูปปากหน่วยเสียง /ฉ,ช,ฌ/ch/
...	...
57	ชื่อรูปปากหน่วยเสียง /เอีย/iia/
58	ชื่อรูปปากหน่วยเสียง /เอือ/vva/
59	ชื่อรูปปากหน่วยเสียง /อัว/uua/
60	ชื่อรูปปากหน่วยเสียง /อำ/am/
61	ชื่อรูปปากหน่วยเสียง /ไ-,ใ-/aj/
62	ชื่อรูปปากหน่วยเสียง /เ-า/aw/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 โครงสร้างข้อมูล

การทำงานของโปรแกรมนี้ใช้ตัวแปรชนิด string หนึ่งตัวสำหรับเก็บค่าชื่อ Blend shape ของโมเดล ใช้ Array ขนาด 61 ของตัวแปรชนิด string ในการเก็บค่าชื่อรูปปากหน่วยเสียงของโมเดล และใช้ตัวแปรชนิด string อีกหนึ่งตัวสำหรับเก็บค่าข้อความที่ผู้ใช้พิมพ์เข้ามา ดังรูปที่ 4.3

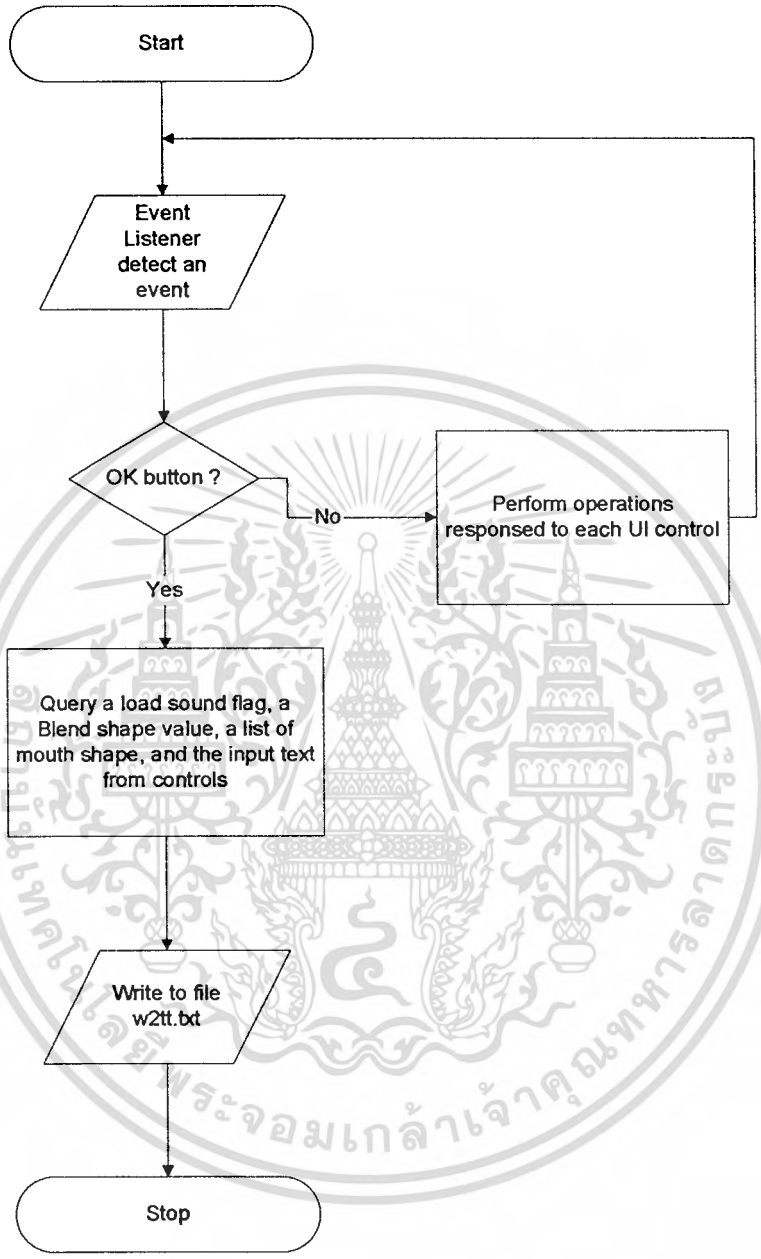
```
string $bShapes;
global string $phoneme[61];
string $input_text;
```

รูปที่ 4.3 โครงสร้างข้อมูลส่วนย่อยติดต่อกับผู้ใช้

4.3.3 อัลกอริธึมโปรแกรมโมดูลย่อยส่วนติดต่อกับผู้ใช้

เมื่อโปรแกรมนี้ถูกเรียก จะมีการสร้างหน้าต่างของโปรแกรมขึ้นมาใหม่ พร้อมกับสร้างส่วนคอนโทรล (Control) ต่างๆ เช่น คอนโทรลชนิด scrollField ชื่อ SCROLLINPUTOBJ สำหรับให้ผู้ใช้พิมพ์ข้อความที่ต้องการทำภาพการเคลื่อนไหว คอนโทรลชนิด Button ชื่อ OK สำหรับสั่งให้ประมวลผล คอนโทรลชนิด textScrollList ชื่อ BS_WEIGHT สำหรับให้ผู้ใช้เลือกรูปปาก คอนโทรล optionMenu ชื่อ W2TTVISEME สำหรับให้เลือกหน่วยเสียงเป็นต้น จากนั้นโปรแกรมจะรอเหตุการณ์ที่เกิดขึ้นกับคอนโทรลต่างๆเหล่านี้และจะตอบสนองต่อเหตุการณ์ที่เกิดขึ้นแตกต่างกันไปตามคอนโทรลและเหตุการณ์ที่เกิดขึ้น รูปที่ 4.4 แสดงลักษณะการทำงานของโปรแกรมนี้ จากรูปเมื่อผู้ใช้คลิ้กคอนโทรลชนิด Button ชื่อ OK จะทำให้มีการประมวลผลข้อมูลจากคอนโทรลอื่นๆ ทั้งหมดแล้วเขียนเป็นไฟล์ w2tt.txt แล้วจบโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Flow chart อัลกอริธึมโปรแกรม want2talkthaiOptions.mel

4.3.3.1 อัลกอริธึมการเรียกข้อมูลจากคอนโทรล

เมื่อผู้ใช้คลิ้กปุ่ม OK ขั้นตอนส่วนนี้จะถูกเรียกเพื่อเตรียมข้อมูลทั้งหมดก่อนจะเขียนลงไฟล์ w2tt.txt ด้วยโพรโตคอลเฉพาะที่กล่าวมาแล้ว อัลกอริธึมส่วนนี้แสดงได้ดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
global string $phoneme[61];

// store sound flag and input text
string $load_sound_flag = getSound();
string $input_text = getTextInput();

// set data to write to file
global proc query_data()
{
  string $cmd_string, $v;
  string $blendShapes[] = getListofSelectedBlendShape();
  int $i, $found = 0, $target_num = 61;
  int $fileId;

  $v = "-v ";
  if($blendShapes[0]!="")  $v = $v + $blendShapes[0]  + " ";
  for($i=0;$i<$target_num;$i++)
  {
    if($phoneme[$i]=="") $phoneme[$i] = "NONE";
    $v = $v + $phoneme[$i] + " ";
  }
 if($load_sound_flag)
 {
    // call with option -s import sound into maya time line
    $cmd_string = "-s " + $v + "-t ";
 }
 else
 {
   $cmd_string = $v + "-t ";
 }

 $result = $cmd_string + $input_text;
}
```

รูปที่ 4.5 อัลกอริธึมการเรียกข้อมูลจากคอนโทรล

จากรูปที่ 4.5 เริ่มต้นที่ตัวแปร $phoneme ขณะนี้มีค่าเป็นชื่อรูปปากหรือไม่ก็เป็นค่าว่าง ตัวแปร $load_sound_flag เก็บค่าสถานะการโหลดไฟล์เสียง ตัวแปร $input_text เก็บค่าข้อมูลที่ผู้ใช้ได้พิมพ์เข้ามา จากนั้นฟังก์ชัน query_data จะสร้างส่วนข้อมูลรูปปากก่อน โดยสร้างรหัสควบคุม –v แล้วตามด้วยชื่อ Blend shape ที่ผู้ใช้เลือกมาจากนั้นจึงตามด้วยข้อมูลตัวแปร $phoneme ทั้ง 61ตัว ตามลำดับโดยข้อมูลทั้ง 61 ตัวนี้ หากตัวไหนมีค่าเป็นว่างให้ใส่ค่าเป็น "NONE" แทน เพื่อให้แน่ใจว่าเป็นค่าว่างจริงๆ เมื่อสร้างส่วนข้อมูล –v แล้ว จึงมาตรวจสอบตัวแปร $load_sound_file เพื่อดูว่าผู้ใช้ต้องการให้โหลดไฟล์เสียงหรือไม่ ถ้าต้องการจะสร้างตัวแปรใหม่ชื่อ $cmd_string โดยให้ค่าเป็น $cmd_string = "-s " + $v + "-t "; แต่ถ้าไม่ต้องการโหลดเสียงให้ค่าเป็น $cmd_string = $v + "-t "; จากนั้นจึงสร้างตัวแปรใหม่ชื่อ $result แล้วเก็บค่า $result = $cmd_string + $input_text ; ซึ่งค่าตัวแปร $result นี้จะเป็นค่าที่จะเขียนลงไฟล์ w2tt.txt ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 โปรแกรมการโหลดเสียง

โปรแกรมนี้ใช้โหลดไฟล์เสียงพูดเข้าสู่ Time line เป็นโปรแกรมแบบ MEL script ชื่อ want2talkthaiLoadAudioFile.mel ซึ่งจะถูกโปรแกรมหลักเรียกใช้หากผู้ใช้กำหนดที่ส่วนติดต่อกับผู้ใช้นี้ โปรแกรมนี้แสดงได้ดังรูปที่ 4.6

```
global proc want2talkthaiLoadAudioFile( string $fileName )
{
 global string $gPlayBackSlider;
 string $audioNode = `sound -file $fileName -offset 0`;
 timeControl -e -ds true -sound $audioNode $gPlayBackSlider;
}
```

รูปที่ 4.6 โปรแกรม want2talkthaiLoadAudioFile.mel

4.5 การเก็บโปรแกรม

โปรแกรมที่เกี่ยวข้องในโมดูลย่อยส่วนติดต่อกับผู้ใช้จะมี 3 โปรแกรมคือ

- want2talkthaiMenu.mel
- want2talkthaiOptions.mel
- want2talkthaiLoadAudioFile.mel

ทั้ง 3 โปรแกรมจะต้องถูกเก็บใน Path ซึ่งโปรแกรมมายาจะสามารถเรียกใช้ได้ ซึ่งโดยค่า Default แล้วคือโฟลเดอร์ C:\Program Files\Alias\Maya7.0\scripts\others

4.6 สรุป

ในส่วนโมดูลย่อยการติดต่อกับผู้ใช้เป็นส่วนที่ถูกพัฒนาโดยใช้ MEL script ประกอบด้วย โปรแกรมที่เกี่ยวกับการสร้างเมนูใหม่ โปรแกรมที่ส่งค่าให้โปรแกรมหลัก และโปรแกรมโหลดไฟล์เสียง สำหรับการส่งข้อมูลให้กับโมดูลหลักนั้นจะส่งในรูปของไฟล์ข้อความ คือ w2tt.txt โดยภายในไฟล์จะมีรหัสควบคุมทำให้โมดูลหลักสามารถรู้ลักษณะของข้อมูลภายในไฟล์ และนำไปใช้ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง

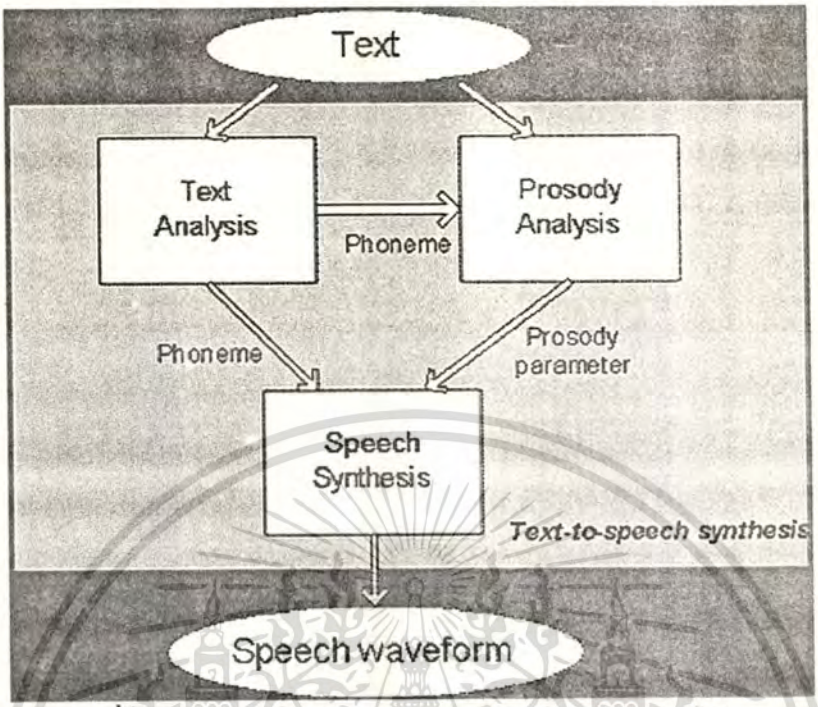
5.1 โปรแกรมวาจา (Vaja) : ซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทย

ในปี พ.ศ. 2541 ทีมนักวิจัยแห่งศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ได้เริ่มวิจัยและพัฒนาโครงการเทคโนโลยีสังเคราะห์เสียงพูดภาษาไทย จนเกิดเป็นผลงานซอฟต์แวร์สังเคราะห์เสียงพูดภาษาไทยขึ้นมาโปรแกรมหนึ่งคือ วาจา (Vaja) ซึ่งมีลักษณะการใช้งานคือให้ผู้ใช้พิมพ์ข้อความภาษาไทยเข้าไป จากนั้นโปรแกรมวาจาจะวิเคราะห์ข้อความนั้นออกเป็นหน่วยเสียงต่างๆ และนำหน่วยเสียงเหล่านั้นมาสังเคราะห์เป็นเสียงพูดออกทางลำโพง ปัจจุบันโครงการนี้ก็ยังคงดำเนินงานอยู่ และโปรแกรมวาจาได้ถูกพัฒนาจนถึงเวอร์ชัน 3.1 ซึ่งมีความสามารถในการเขียนไฟล์หน่วยเสียงซึ่งเป็น Text File และไฟล์เสียงซึ่งเป็น .wav File ได้ นอกจากนี้ ทีมนักวิจัยยังได้พัฒนา Vaja API ซึ่งเป็น C++ API ซึ่งมีลักษณะเป็นไลบรารีสำหรับใช้ในการเข้าถึงระบบภายในโปรแกรมวาจา ดังนั้นเราจึงสามารถนำความสามารถของโปรแกรมวาจามาใช้งานในระบบของเราได้ สำหรับโปรแกรมวาจาจะสามารถดาวน์โหลด (Download) ได้ที่เว็บไซต์ http://vaja.nectec.or.th/VajaTTS_Th.html

5.2 โครงสร้างของระบบสังเคราะห์เสียง

โครงสร้างของระบบสังเคราะห์เสียงโดยทั่วไป สามารถแบ่งการทำงาน ภายใน ได้เป็น 3 ส่วนดังรูปที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 Block diagram ของโครงสร้างระบบสังเคราะห์เสียง

5.2.1 ส่วนการวิเคราะห์ข้อความ (Text Analysis)

ส่วนนี้จะมีหน้าที่วิเคราะห์ข้อความอินพุทเพื่อแปลงเป็นข้อมูล เสียงอ่าน (phoneme) ของคำนั้น และส่งต่อให้ส่วนของการสังเคราะห์เสียง (Speech synthesis) ต่อไป นอกจากนี้ส่วนนี้ยังทำหน้าที่อย่างอื่น ได้แก่

- การแบ่งประโยคจากข้อความที่ยาว (Sentence breaking)
- การทำข้อความให้อยู่ในรูปปกติ (Text Normalization) ได้แก่ การแปลงตัวเลข คำย่อ และเครื่องหมายอื่นๆ ที่ไม่ใช้ข้อ ความให้ กลายเป็นข้อความ
- การหาขอบเขตของวลีของการอ่านในประโยค

5.2.2 ส่วนการวิเคราะห์สัทสัมพันธ์ (Prosody Analysis)

ส่วนนี้ทำหน้าที่ในการวิเคราะห์และสังเคราะห์ข้อมูล สัทสัมพันธ์ (prosody) ของประโยคใดๆ จากข้อมูลเสียงอ่าน และข้อความ ข้อมูลสัทสัมพันธ์ ที่วิเคราะห์ ออกมาได้ในระบบทั่วไปได้แก่

- Segment Duration คือความยาวของเสียงย่อยที่ต้องการสังเคราะห์ ค่านี้จะ มีผลต่อ จังหวะของเสียงที่ทำการสังเคราะห์ เช่น ถ้ากำหนดให้ค่าความยาวของเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย่อยที่ต้องการสังเคราะห์มีขนาดสั้น เสียงที่ทำการสังเคราะห์ก็จะเหมือนกับการพูดเร็ว

- Pitch Contour คือ ค่าความสัมพันธ์ของความถี่มูลฐานกับเวลา ค่านี้จะมีผลต่อเสียงสูงต่ำ (intonation) ของประโยคนั้นๆ

5.2.3 ส่วนการสังเคราะห์เสียงพูด (Speech synthesis)

ส่วนนี้ทำหน้าที่ในการสร้างสัญญาณคลื่นเสียงจากข้อมูลเสียงอ่าน (phonetic transcription) และข้อมูลสัทสัมพันธ์ (Prosody transcription) จากข้อ 5.2.1 และ 5.2.2 และส่งออกสู่ลำโพง เพื่อให้เราได้ยินเสียงพูดประโยคนั้นๆ โดยทั่วไปส่วนนี้สามารถแบ่งตามเทคนิควิธีการสังเคราะห์เสียง ได้ 3 ประเภท คือ

- Formant Synthesis

เทคนิคการสังเคราะห์วิธีแบบนี้ ข้อมูลเสียงอ่านใดๆ จะถูกกำหนดไว้อยู่ในรูปของความถี่ฟอร์แมนท์ต่างๆ (F1, F2, F3) ของเสียงนั้นๆ เมื่อต้องการสังเคราะห์เสียงใดๆ ก็นำข้อมูลเหล่านี้มาทำการสังเคราะห์ให้เป็นสัญญาณเสียง ซึ่งวิธีการนี้จะมีข้อดีที่สามารถควบคุมค่าความเปลี่ยนแปลงของ ความถี่ฟอร์แมนท์ (Formant transition) ที่บริเวณรอยต่อระหว่างเสียงได้ง่าย แต่มีข้อเสีย คือ การจะแทนเสียงใดๆ ด้วยค่าฟอร์แมนท์ทำได้ยาก จะต้องมีกฎในการสังเคราะห์เสียงใดๆ จำนวนมาก และเสียงที่สังเคราะห์ออกมาได้จะไม่ค่อยเป็นธรรมชาติ ตัวอย่างของระบบแบบนี้ได้แก่ MITALK และ DECTALK

- Articulation Synthesis

สำหรับวิธีนี้ข้อมูลเสียงที่ต้องการสังเคราะห์จะโมเดลอยู่ในรูปของค่าพารามิเตอร์ของโครงสร้างทางกายภาพของการเคลื่อนไหวของอวัยวะในช่องปากที่ทำให้เกิดเสียงต่างๆ วิธีการนี้ค่อนข้างยากในแง่การโมเดล เสียงต่างๆ ซึ่งจะต้องศึกษาจากอวัยวะในการออกเสียงจริงๆ

- Concatenation synthesis

เทคนิควิธีการนี้เป็นวิธีที่ได้รับความนิยมมากที่สุดในปัจจุบัน โดยเสียงที่ทำการสังเคราะห์ขึ้นนี้เกิดจากการนำหน่วยเสียงย่อยที่ทำการเก็บไว้ก่อน แล้วมาต่อกันเป็นเสียงพูดที่ต้องการ โดยทั่วไปหน่วยเสียงย่อยที่ทำเก็บไว้จะอยู่ระดับต่ำกว่าคำ เช่น หน่วยของเสียงพยางค์ หน่วยของเสียงครึ่งพยางค์ (demisyllable) หน่วยของเสียงคู่เสียง (diphone) เป็นต้น ซึ่งในโปรแกรมวาจานี้ได้เลือกใช้เทคนิควิธีการนี้ โดยใช้หน่วยของ เสียงครึ่งพยางค์ เป็นพื้นฐาน

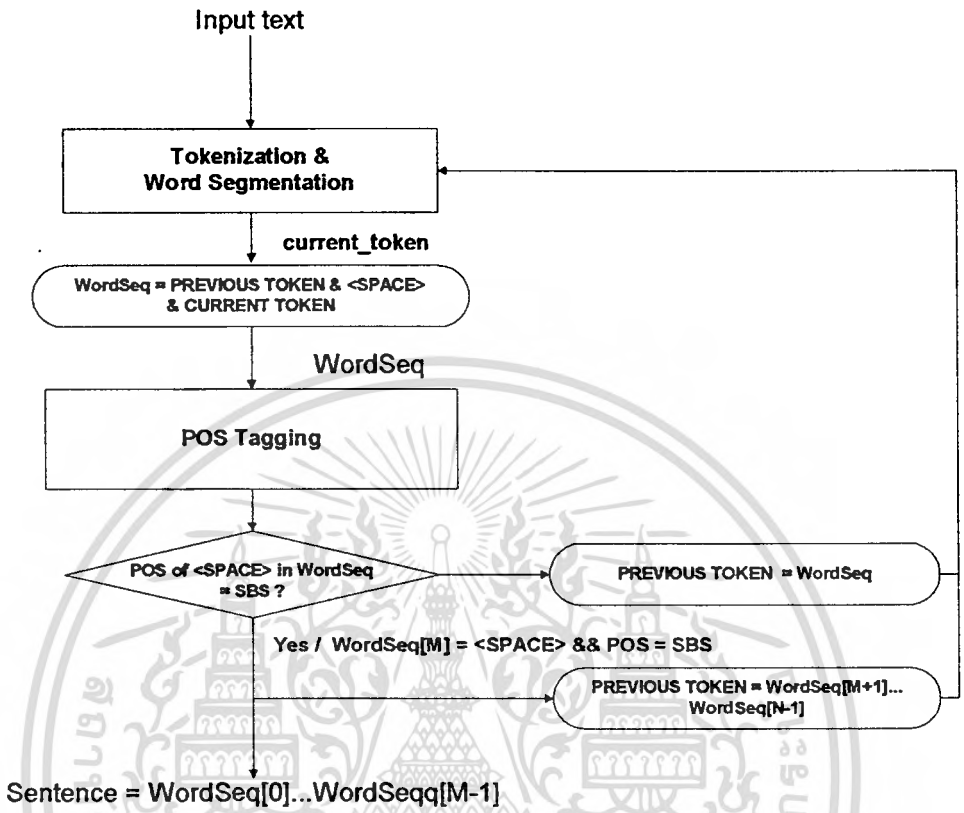
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การวิเคราะห์ข้อความ (Text Analysis) ของ Vaja API

การวิเคราะห์ข้อความคือการรับข้อความที่ถูกพิมพ์เข้าสู่ระบบย่อยการแปลงข้อความเป็นหน่วยเสียงของ Vaja API ในทางปฏิบัติแล้วข้อความที่ถูกพิมพ์เข้ามาสู่ระบบอาจเป็นเพียงหนึ่งย่อหน้าหรือหลายย่อหน้า แต่ละย่อหน้าประกอบด้วยประโยคหลายๆ ประโยค ข้อความนั้นอาจจะรวมคำภาษาไทยและสัญลักษณ์อื่นๆ เช่น ตัวเลข ตัวย่อ เป็นต้น แต่เพราะว่าจุดประสงค์ของโปรแกรมวาจาคือเพื่อสังเคราะห์เสียงพูดของข้อความภาษาไทย ดังนั้นข้อความภาษาต่างประเทศที่ปรากฏอยู่ในข้อความจะถูกเพิกเฉยไป การวิเคราะห์ข้อความจะแบ่งส่วนข้อความนำเข้าลงเป็นหน่วยเล็กๆเพื่อสะดวกสำหรับการประมวลผล หน่วยเล็กๆ ที่ได้นี้คือลำดับของประโยคและวลี (Phrase) นอกจากการกำหนดส่วนที่เป็นประโยคและวลีแล้ว ยังมีฟังก์ชันที่เรียกว่า "Graphme-to-phoneme" ซึ่งทำหน้าที่แปลงจากข้อความเป็นหน่วยเสียงด้วย

อัลกอริธึม (Algorithm) สำหรับการแยกคำในประโยคนั้น ไม่เหมือนภาษาอังกฤษเพราะว่าในภาษาไทยไม่มีเครื่องหมายเว้นวรรคที่แสดงการแบ่งคำแต่ละคำ เครื่องหมายเว้นวรรคจะใช้เมื่อจบประโยคภาษาไทย แต่ก็ไม่จำเป็นเสมอไปที่เครื่องหมายเว้นวรรคในหนึ่งย่อหน้าจะแสดงถึงการจบประโยคทุกครั้ง เครื่องหมายเว้นวรรคนี้ยังสามารถถูกใช้ในความหมายอื่นได้อีกเช่น ใช้ระหว่างวลีหรืออนุประโยค (Clause) ภายในประโยค ก่อนและหลังตัวเลข เป็นต้น ดังนั้นจึงแบ่งเครื่องหมายเว้นวรรคเป็น 2 แบบ คือ แบบจบประโยค และแบบไม่จบประโยค อัลกอริธึมนี้แสดงได้ดังรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 Block diagram ของการแยกประโยคภาษาไทย

กระบวนการ Tokenization & Word Segmentation จะแยกชุดของโทเคนต่างๆ (a set of tokens) ด้วยเครื่องหมายเว้นวรรคหนึ่งอันเป็นอย่างน้อยแทรกระหว่างชุดของโทเคนนั้น เครื่องหมายเว้นวรรคทั้งหลายที่อยู่ในชุดของโทเคนจะถูกแบ่งว่าเป็นแบบไหนโดย POS tagging ความหมายของโทเคนหนึ่ง คือ ตัวอักษรหลายตัวติดกันที่ล้อมด้วยเครื่องหมายเว้นวรรค สำหรับเครื่องหมายพิเศษ เช่น ตัวเลข จะถูกพิจารณาเป็นกรณีพิเศษ เช่น 10600 สามารถออกเสียงได้ 2 แบบ คือ หนึ่งศูนย์หกศูนย์ศูนย์ หรือ หนึ่งหมื่นหกร้อย ก็ได้แล้วแต่ใช้ในกรณีแบบไหนซึ่งเป็นหน้าที่ของการทำ Normalization ซึ่งรวมอยู่ในกระบวนการ Tokenization และ Grapheme-to-phoneme Conversion กระบวนการ Tokenization ทำหน้าที่เปลี่ยนตัวเลขให้กลายเป็นหนึ่งโทเคน จากนั้น Grapheme-to-phoneme Conversion จะกำหนดหน่วยเสียงให้ตามลักษณะของข้อความนั้น ยิ่งไปกว่านั้นโทเคนหนึ่งจะถูกแบ่งเป็นหลายคำเพราะว่าไม่มีเครื่องหมายแยกคำที่ชัดเจนในภาษาไทย

โทเคนที่ติดกัน 2 อันจะถูกรวมกันเป็น Word Sequence โดยมีเครื่องหมายเว้นวรรคแทรกตรงกลางหนึ่งอัน สำหรับเครื่องหมายเว้นวรรคหลายๆ อันใน Word Sequence จะถูกแบ่งว่าเป็นแบบไหนระหว่างจบประโยค (Sentence Break Space) หรือแบบไม่จบประโยค (Non Sentence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Break Space) โปรแกรมวาจานิยามปัญหาการแบ่งนี้อยู่ในเทอมของ Statistical POS tagging กล่าวคือ ลำดับที่น่าจะเป็นไปได้มากที่สุดของ POSs หลายๆ อันและการมอบหมาย Individual Word-level POS เป็นสิ่งที่จะใช้กำหนดให้ว่า POS อันไหนที่น่าจะเป็นไปได้มากที่สุด ดังนั้นการแบ่งประเภทนี้จึงมีหน้าที่ในการกำหนดว่า POS ของเครื่องหมายเว้นวรรคหลายอันในลำดับที่น่าจะเป็นไปได้มากที่สุดของ POS เป็นการปิดประโยคหรือไม่ ซึ่งใช้ 2 อย่างคือการคำนวณความน่าจะเป็นของลำดับ POS และการคำนวณลำดับของ POS ที่น่าจะเป็นไปได้มากที่สุด

เนื่องจากเป็นไปได้ว่าเครื่องหมายเว้นวรรคใน Word Sequence ซึ่งถูกใช้เป็นแบบไม่ปิดประโยค (Non Sentence Break Spaces) นั้นจะถูกให้ค่าที่ไม่ถูกต้อง ดังนั้นจึงต้องตรวจหาเครื่องหมายเว้นวรรคระหว่างโทเคนปัจจุบัน (current token) และโทเคนก่อนหน้า (previous tokens) และเครื่องหมายเว้นวรรคทั้งหลายภายในโทเคนก่อนหน้า ถ้าไม่มีเครื่องหมายเว้นวรรคแบบจบประโยคแล้ว ทุก Word Sequence นี้จะถูกใช้เป็นโทเคนก่อนหน้าในการทำซ้ำครั้งต่อไป (Next iteration) แต่ถ้าพบเครื่องหมายเว้นวรรคแบบจบประโยคแล้ว ประโยคผลลัพธ์ที่ได้จากกระบวนการวิเคราะห์ข้อความก็คือลำดับทั้งหมดจากคำแรกถึงคำที่อยู่ก่อนเครื่องหมายเว้นวรรคแบบจบประโยค ส่วนลำดับของคำที่เหลือซึ่งอยู่หลังจากเครื่องหมายเว้นวรรคแบบจบประโยคนี้จะถูกใช้เป็นโทเคนแรกในการทำซ้ำครั้งต่อไป อัลกอริธึมนี้จะแยกลำดับของโทเคนต่างๆ และตรวจชนิดของเครื่องหมายเว้นวรรคที่อยู่ระหว่างโทเคนจนกระทั่งหมดย่อหน้า สังเกตได้ว่าอัลกอริธึมนี้สามารถแก้ปัญหาความจำกัดของทรัพยากรหน่วยความจำและเวลาในการประมวลผล เพราะว่าจะประมวลผลทีละโทเคนมากกว่าจะทำพร้อมกันหมดทั้งย่อหน้าในเวลาเดียว

5.3.1 การกำหนดขอบเขตวลี (Phrase boundary determination)

โดยปกติเมื่อเราพูดประโยคยาวๆ หนึ่งประโยค เรามักจะตัดประโยคลงเป็นวลี จากงานวิจัยของ Luksaneeyanawin เรื่องการศึกษาตัวอักษรและฟังก์ชันของการหยุด (The study of charactersics and function of pause) ทำให้ทราบว่าตำแหน่งของการหยุดเกิดขึ้นเฉลี่ยที่ทุกๆ 8 พยางค์ (S.D. = 4) ขอบเขตวลีเหล่านี้จับ (capture) ตัวอักษร Prosodic หลายอย่างเช่น การหยุดที่จุดสิ้นสุดของวลี, ถ้าขอบเขตของวลีนั้นถูกกำหนดอย่างถูกต้องแล้ว การสร้างสัทสัมพันธ์จะเล่นบทบาทสำคัญในการรับเสียงพูดธรรมชาติมากกว่า และอัลกอริธึมสำหรับการกำหนดขอบเขตวลีใช้ข้อมูลออก (Output) ของกระบวนการแยกประโยค โดยการกำหนดครั้งแรกอย่างคร่าวๆที่ตำแหน่งตัดวลีทั้งหลายในประโยค จากนั้นจึงรวมตำแหน่งเหล่านี้เข้าด้วยกันโดยการใช้การควบคุมจำนวนพยางค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริธึมนี้จะใส่ตัวตัดวลีอย่างคร่าวๆ ตัวหนึ่งที่ตำแหน่งของเครื่องหมายเว้นวรรค (Space) และเครื่องหมายวรรคตอน (Punctuation mark) อัลกอริธึมนี้จะตรวจจับเครื่องหมายเว้นวรรคตัดวลีโดยใช้กฎเกณฑ์เฉพาะที่มาจากรูปแบบการเขียนภาษาไทยอย่างเป็นทางการ ส่วนตำแหน่งตัดวลีคร่าวๆ อื่นๆ จะถูกกำหนดโดยการตรวจหาแต่ละคู่ของคำจากซ้ายไปขวาในประโยค กฎเกี่ยวกับคำที่เป็นเนื้อหาและคำที่เป็นฟังก์ชันจะถูกใช้เพื่อใส่ตำแหน่งตัดเบื้องต้นก่อนคำที่เป็นฟังก์ชันทุกคำซึ่งตามหลังคำที่เป็นเนื้อหา และคำที่เป็นฟังก์ชันก็คือคำที่เป็นสันธาน (Conjunction) , คำบุรพบท (Preposition), และคำสรรพนาม (Relative pronoun) ส่วนคำที่เป็นเนื้อหาก็คือคำที่เหลือทั้งหมดซึ่งไม่ตรงกับคำที่เป็นฟังก์ชัน ยิ่งไปกว่านี้ กฎที่มาจากงานวิจัยของ Luksaneeyanawin ยังให้ตำแหน่งตัดก่อนหรือหลังบางคำที่เป็นไวยากรณ์พิเศษ

ในขั้นตอนการรวมตำแหน่งตัดวลีอย่างคร่าวทั้งหลายในประโยค ตำแหน่งตัดเหล่านี้จะถูกรวมเป็นวลีเดียว ถ้าตำแหน่งเหล่านี้ไม่สิ้นสุดในเครื่องหมายเว้นวรรคแบบตัดวลีหรือเครื่องหมายวรรคตอน และบรรจุพยางค์ 10 หรือน้อยกว่า พยางค์เหล่านี้ถูกรวมเข้ากับวลีที่ตามมาจนกระทั่งเครื่องหมายวรรคตอนหรือเครื่องหมายเว้นวรรคแบบตัดวลีถูกพบหรือจนกระทั่งจำนวนของพยางค์มากกว่า 10 พยางค์

5.3.2 การแปลงข้อความเป็นหน่วยเสียง (Grapheme-to-phoneme conversion)

ข้อมูลออกทั้งหลายจากกระบวนการแยกประโยคและการกำหนดขอบเขตวลีคือหนึ่งประโยคที่มีตำแหน่งตัดวลีและมีขอบเขตของคำ ในขั้นตอนนี้การแสดงหน่วยเสียงของแต่ละคำจะถูกกำหนด แต่ละคำแรกจะถูกให้หน่วยเสียงโดยหาจากพจนานุกรมการออกเสียง (Pronunciation dictionary) ซึ่งมีเสียงอยู่ 25,000 เสียง และพัฒนาเป็นกฎตัวอักษรเป็นเสียง (Letter-to-sound rule) เพื่อใช้จัดการคำที่ไม่อยู่ในพจนานุกรม กฎนี้ประกอบด้วย 2 ขั้นตอน ขั้นตอนแรก Grapheme ของคำหนึ่งจะถูกแบ่งเป็นลำดับของพยางค์หนึ่งและขั้นตอนที่สอง ลำดับของพยางค์ที่ได้จะถูกแปลงเป็นลำดับของหน่วยเสียง (Phoneme string) โดยการจับคู่แบบ Orthographical-to-phonological mapping สำหรับโทนเสียง (Tone) ของแต่ละพยางค์จะถูกให้โดยการพิจารณาองค์ประกอบหน่วยเสียง (พยัญชนะต้น สระ ตัวสะกด) และเครื่องหมายกำกับหน่วยเสียงทาง Orthographical ของมัน

ขั้นตอนการกำหนดพยางค์ (Syllabification) จะให้ข้อมูลออกเป็นองค์ประกอบทางพยางค์ คือ พยัญชนะต้น สระ ตัวสะกด และโทนเสียง ข้อดีของโครงสร้างนี้คือยืดหยุ่นในการปรับปรุงและประมวลผลได้อย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 การสร้างสัทสัมพันธ์ (Prosody Generation)

สัทสัมพันธ์ (Prosody) หมายถึงคุณสมบัติของเสียงพูด เช่น การเปลี่ยนแปลงระดับเสียง (Pitch variation) ความดัง (Loudness) และความยาวพยางค์ (Syllable length) ผลกระทบของสัทสัมพันธ์ถูกอ้างเป็นปรากฏการณ์เกี่ยวกับลักษณะการออกเสียง เนื่องจากมันเกิดขึ้นในระดับสูงกว่าเสียงพูดธรรมดา เช่น พยางค์หรือหน่วยเสียง ความเป็นธรรมชาติของเสียงสังเคราะห์จะถูกกระทบโดยสัทสัมพันธ์

5.4 การสังเคราะห์เสียงพูด (Speech Synthesis)

หลังจากพารามิเตอร์ (Parameters) ที่จำเป็นทั้งหมดสำหรับการสังเคราะห์ถูกกำหนด ส่วนนี้จะใช้พารามิเตอร์เหล่านี้เพื่อกำหนดว่าหน่วยเสียงไหนควรถูกเลือกและหน่วยเสียงเหล่านี้ควรถูกประมวลผลเพื่อสังเคราะห์เสียงพูดอย่างไร พารามิเตอร์ทั้งหลายสามารถถูกแบ่งได้ 2 กลุ่ม กลุ่มแรกถูกสร้างจากกระบวนการวิเคราะห์ข้อความ (Text analysis) ซึ่งประกอบด้วยรายการเสียงของวลี รายการเหล่านี้ถูกใช้เพื่อเลือกหน่วยเสียงที่เกี่ยวข้อง ส่วนอีกกลุ่มถูกสร้างจากกระบวนการสร้างสัทสัมพันธ์ (Prosody Generation) ซึ่งประกอบด้วยช่วงเวลาและ F0 Contours ทั้งหมดจะถูกใช้ในการประมวลผลสัญญาณ (Signal Processing) เพื่อปรับปรุงความเป็นธรรมชาติของเสียงสังเคราะห์

5.4.1 แบบแผนการสังเคราะห์ (Synthesis Scheme)

มีหลายชนิดของหน่วยการสังเคราะห์ที่ถูกใช้ในการสังเคราะห์เสียงพูดแบบ Concatenative เช่น คำ (Words) พยางค์ (Syllables) ครึ่งพยางค์ (Demisyllables) หน่วยเสียง (Phonemes) คู่เสียง (Diphones) สามเสียง (Triphones) เป็นต้น แต่ละชนิดมีข้อดีต่างกันขึ้นอยู่กับจุดประสงค์ของแต่ละระบบ สำหรับ Vaja API จะใช้ครึ่งพยางค์เพราะว่าเป็นจำนวนของหน่วยเสียงที่กำลังพอดีและคุณภาพที่ยอมรับได้ ถึงแม้ว่าคุณภาพเสียงของมันที่ขอบเขตของพยางค์ไม่ค่อยเป็นธรรมชาติเหมือนเสียงพูดจริงแต่ปัญหานี้สามารถถูกปรับปรุงได้โดยการประมวลผลสัญญาณ (Signal Processing) ที่นำมาใช้ในระบบ

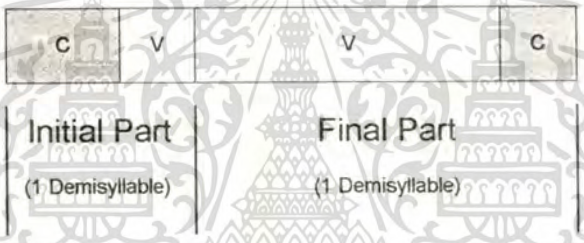
เสียงครึ่งพยางค์คือหน่วยเสียงที่เป็นครึ่งต้นและครึ่งท้ายของพยางค์ แนวคิดนี้คือรูปคลื่นของเสียงพูด (Speech waveform) จะถูกสร้างโดยการต่อกันของส่วนที่เป็นพยางค์ รูปคลื่นของพยางค์หนึ่งถูกสร้างจากหน่วยครึ่งพยางค์ต้นและหน่วยครึ่งพยางค์ท้าย หน่วยทั้งสองถูกแบ่งจากพยางค์หนึ่งที่ส่วนสระ โดยทั่วไปแล้ว พยางค์ภาษาไทยมีโครงสร้างเป็น CVC ดังนั้นพยางค์ถูกแบ่งเป็น 2 ส่วนคือ CV และ VC หน่วยต้นบรรจุพยัญชนะเดี่ยว (Single consonants) หรือพยัญชนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบกล้ำ (Double consonants) เชื่อมกับสระหนึ่ง หน่วยส่วนท้ายบรรจุสระหนึ่งเชื่อมกับตัวสะกด และลักษณะของโทนเสียงด้วย

5.4.1.1 โครงสร้างหน่วยครึ่งพยางค์ (Demisyllable unit structure)

เสียงพยางค์ภาษาไทยหนึ่งมี 4 องค์ประกอบ คือ พยัญชนะต้น สระ ตัวสะกด และโทนเสียง การพิจารณาโครงสร้างพยางค์ภาษาไทยมี 4 รูปแบบ คือ CV, CCV, CVC, และ CCVC ตัวอักษร C แสดงพยัญชนะต้น ขณะที่ CC แสดงพยัญชนะควบกล้ำ ตัวอักษร V แสดงถึงสระ และ C ตัวสุดท้ายแสดงตัวสะกด ส่วนเลข 0 – 4 แสดงโทนเสียงวรรณยุกต์ สามัญ เอก โท ตรี และจัตวา ตามลำดับ รูปที่ 5.3 แสดงโครงสร้างของหน่วยครึ่งพยางค์



รูปที่ 5.3 โครงสร้างหน่วยครึ่งพยางค์

พยัญชนะต้นมี 44 ตัวอักษรในภาษาไทย และมี 21 หน่วยเสียง (Phonemes) ซึ่งถูกจัดประเภทเป็น 3 เสียง คือ อักษรเสียงสูง อักษรเสียงกลาง และอักษรเสียงต่ำ การจัดประเภทนี้เป็นสิ่งสำคัญในการกำหนดโทนเสียงของพยางค์ ทุกตัวอักษรสามารถเกิดขึ้นในตำแหน่งเริ่มต้น เช่น กาด [ka;t], บน [bon], โยน [jo;n] เป็นต้น บางหน่วยเสียงสามารถควบกล้ำกันแล้วได้หน่วยเสียงใหม่ เช่น ปล- /pl/, คว- /khw/, ตร- /tr/ เป็นต้น ตารางที่ 1 แสดงพยัญชนะในภาษาไทย

สระมี 2 ประเภทคือสระเดี่ยว (Monophthong) และสระผสม (Diphthong) สระเดี่ยวมี 18 หน่วยเสียงสามารถแบ่งได้เป็น 2 แบบ คือ สระเสียงสั้นและสระเสียงยาวอย่างละ 9 หน่วยเสียง สระเดี่ยวเสียงสั้น เช่น –ะ /a/, อุ /u/ เป็นต้น ส่วนสระเดี่ยวเสียงยาว เช่น –า /a;/, อู /u;/ เป็นต้น ส่วนสระผสมประกอบด้วย 6 หน่วยเสียง เป็นแบบสระเสียงสั้นและสระเสียงยาวอย่างละ 3 เสียง สระผสมเสียงสั้น เช่น เอียะ /ia/, อัวะ /ua/ เป็นต้น ส่วนสระผสมเสียงยาว เช่น เอีย /i;a/, อัว /u;a/ เป็นต้น ตารางที่ 2 แสดงสระในภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวสะกดหรือพยัญชนะท้าย ประกอบด้วย 9 หน่วยเสียง เช่น ราด [ra;t1], กับ [kap1], จะ [caz1] ส่วนโทนเสียง มี 5 หน่วยเสียง ซึ่งเป็นเสียงวรรณยุกต์ทั้ง 5 แทนด้วยตัวเลข 0 – 4 ตามลำดับ โครงสร้างพยางค์และเสียงพยัญชนะต้นสำคัญมากที่จะให้โทนเสียงสำหรับแต่ละพยางค์

5.5 การออกแบบ Class โมดูลย่อยการแปลงข้อความเป็นหน่วยเสียง

เนื่องจากการแปลงข้อความเป็นหน่วยเสียงต้องอาศัย Vaja API ซึ่งอยู่ในรูปของไลบรารี ดังนั้นก่อนอื่นจึงจำเป็นต้องติดตั้งโปรแกรมวาจา (Vaja) ลงในเครื่องก่อนด้วย จากนั้นจึงจะสามารถใช้งานไลบรารีของวาจาได้ สำหรับวิธีการเรียกใช้งานไลบรารีวาจานี้ จะทำโดยการสร้างคลาสขึ้นมาเพื่อทำงานในส่วนของการติดต่อกับไลบรารีวาจาโดยเฉพาะ คือ CVajaTTS ซึ่งคลาสนี้จะทำหน้าที่รับข้อมูลที่เป็นข้อความเข้ามาแล้วแปลงเป็นหน่วยเสียงเขียนลงไฟล์ z.pho และเป็นไฟล์เสียง คือ z.wav ซึ่งไฟล์ทั้งสองนี้จะถูกใช้ในระบบต่อไป รูปที่ 5.4 แสดงคลาส CVajaTTS

```
class CVajaTTS
{
public:
        CVajaTTS();     // standard constructor
        ~CVajaTTS();
        ...
        void saytext(char *);
        void saytexttofile(char *);
        int getstatus();

protected:
        ...
};
```

รูปที่ 5.4 แสดงคลาส CVajaTTS

คลาส CVajaTTS ประกอบด้วย 3 Methods ดังนี้

- saytext(char *) มีหน้าที่แปลงข้อความเป็นไฟล์หน่วยเสียงชื่อ z.pho
- saytexttofile(char *) มีหน้าที่แปลงข้อความเป็นไฟล์เสียงชื่อ z.wav
- getstatus() มีหน้าที่ส่งค่าสถานะการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 โครงสร้างข้อมูลของไฟล์หน่วยเสียง

โครงสร้างของไฟล์หน่วยเสียงที่ได้จาก Vaja API สร้างขึ้นมีรูปแบบเป็นรวมกลุ่มของคำ ซึ่งใช้ตัวอักษรเครื่องหมาย "*" แสดงถึงการสิ้นสุด 1 คำ ภายในคำ ประกอบด้วยหลายพยางค์ซึ่งใช้เครื่องหมาย "|" แสดงถึงการสิ้นสุด 1 พยางค์ และภายในพยางค์ประกอบด้วยพยัญชนะต้น, สระ, ตัวสะกด, และโทนเสียงโดยใช้ตัวอักษร "-" คั่นกลางระหว่างองค์ประกอบเหล่านี้ และใช้เครื่องหมาย "^" ตามหลังพยัญชนะที่เป็นตัวสะกด สำหรับการเว้นวรรคจะแทนด้วยค่า CR/LF ซึ่งการเว้นวรรคนี้จะถูกตีความว่าเป็นการสิ้นสุด 1 คำด้วย ดังนั้นก่อนค่า CR/LF จึงมีอักษร "*" เสมอ

ตัวอย่างการแทนข้อมูลในไฟล์สำหรับข้อความ "คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง" เป็นดังรูปที่ 5.5

```
6B 68 2D 61 2D 7A 5E 2D 33 7C 6E 2D 61 2D 7A 5E ; kh-a-z^-3|n-a-z^
2D 33 7C 2A 74 68 2D 65 2D 6B 5E 2D 33 7C 6E 2D ; -3|*th-e-k^-3|n-
6F 6F 2D 7A 5E 2D 30 7C 6C 2D 6F 6F 2D 7A 5E 2D ; oo-z^-0|l-oo-z^-
30 7C 6A 2D 69 69 2D 7A 5E 2D 30 7C 2A 73 2D 61 ; 0|j-ii-z^-0|*s-a
61 2D 7A 5E 2D 34 7C 72 2D 61 2D 7A 5E 2D 33 7C ; a-z^-4|r-a-z^-3|
73 2D 6F 2D 6E 5E 2D 34 7C 74 68 2D 65 65 2D 74 ; s-o-n^-4|th-ee-t
5E 2D 32 7C 2A 0D 0A 73 2D 61 2D 7A 5E 2D 31 7C ; ^-2|*..s-a-z^-1|
74 68 2D 61 61 2D 7A 5E 2D 34 7C 62 2D 61 2D 6E ; th-aa-z^-4|b-a-n
5E 2D 30 7C 2A 74 68 2D 65 2D 6B 5E 2D 33 7C 6E ; ^-0|*th-e-k^-3|n
2D 6F 6F 2D 7A 5E 2D 30 7C 6C 2D 6F 6F 2D 7A 5E ; -oo-z^-0|l-oo-z^
2D 30 7C 6A 2D 69 69 2D 7A 5E 2D 30 7C 2A 70 68 ; -0|j-ii-z^-0|*ph
72 2D 61 2D 7A 5E 2D 33 7C 63 2D 40 40 2D 6D 5E ; r-a-z^-3|c-@@-m^
2D 30 7C 2A 6B 6C 2D 61 61 2D 77 5E 2D 32 7C 2A ; -0|*kl-aa-w^-2|*
0D 0A 63 2D 61 61 2D 77 5E 2D 32 7C 6B 68 2D 75 ; ..c-aa-w^-2|kh-u
2D 6E 5E 2D 30 7C 2A 74 68 2D 61 2D 7A 5E 2D 33 ; -n^-0|*th-a-z^-3
7C 68 2D 61 61 2D 6E 5E 2D 34 7C 2A 6C 2D 61 61 ; |h-aa-n^-4|*l-aa
2D 74 5E 2D 32 7C 2A 6B 72 2D 61 2D 7A 5E 2D 31 ; -t^-2|*kr-a-z^-1
7C 2A 62 2D 61 2D 6E 67 5E 2D 30 7C 2A 0D 0A    ; |*b-a-ng^-0|*..
```

รูปที่ 5.5 ข้อมูลไฟล์หน่วยเสียงสำหรับข้อความ "คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง"

5.7 สรุป

โมดูลย่อยการแปลงข้อความเป็นหน่วยเสียงถูกสร้างเป็น CVajaTTS class ซึ่งมี 2 Methods ที่ทำหน้าที่แปลงข้อความเป็นไฟล์หน่วยเสียงและไฟล์เสียงพูด โดยใช้ฟังก์ชันจากไลบรารี Vaja API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
โมดูลย่อยการสร้างรูปปากและโมดูลย่อยสร้างการเคลื่อนไหว

6. โมดูลย่อยการสร้างรูปปาก

โมดูลย่อยการสร้างรูปปากมี 3 หน้าที่หลักคือ 1) สร้างรูปปากสำหรับแต่ละหน่วยเสียงภาษาไทย 2) สร้างรูปปากสำหรับแต่ละเสียงพยางค์ 3) สร้างส่วนควบคุมการผสมรูปปาก หรือ Blend shape deformer สำหรับแต่ละรูปปากของหน่วยเสียง

6.1 การสร้างรูปปากสำหรับแต่ละหน่วยเสียงภาษาไทย

ในระบบภาษาไทยประกอบด้วย หน่วยเสียงพยัญชนะทั้งพยัญชนะเดี่ยวและพยัญชนะควบกล้ำ 33 หน่วยเสียง หน่วยเสียงสระทั้งสระเดี่ยว สระคู่ และสระอักษร 27 หน่วยเสียง รวมเป็น 60 หน่วยเสียง ดังตารางที่ 6.1 ตารางที่ 6.2 และตารางที่ 6.3 ตามลำดับ แต่ทั้ง 60 หน่วยเสียงนี้หากนำมาใช้ในการสร้างภาพเคลื่อนไหวของปากทั้งหมดจะทำให้ใช้ทรัพยากรของระบบอย่างมาก อีกทั้งยังใช้เวลามากด้วย ดังนั้นวิธีการแก้ปัญหาจำนวนหน่วยเสียงที่มากเช่นนี้จึงต้องใช้การจัดกลุ่มของหน่วยเสียงที่มีลักษณะรูปปากที่คล้ายกันให้อยู่ในกลุ่มเดียวกัน เรียกว่า Viseme

ในโครงการนี้จะแบ่ง Viseme โดยยึดหลักของอวัยวะที่ใช้ในการออกเสียงและลักษณะการออกเสียง อันได้แก่ เสียงหยุด เสียงเสียดแทรก และเสียงก้อง ทำให้แบ่งได้เป็น 8 กลุ่มดังนี้

1) ริมฝีปาก เสียงหยุดและเสียงก้อง ได้แก่ ม, บ, ป, พ, ภ, ผ, ปร-, ปล-, พร-, ผร-, พล-, ผล-

2) ริมฝีปาก เสียงเสียดแทรกและเสียงก้อง ได้แก่ ฝ, ฟ, ว

3) ฟัน เสียงหยุด ได้แก่ ค, ฎ, ต, ฏ, ถ, ฐ, ท, ฒ, ท, ธ, ตร-, ทร-

4) ฟัน เสียงเสียดแทรก ได้แก่ ส, ศ, ษ, ช

5) ฟัน เสียงก้อง ได้แก่ น, ณ, ล, ฬ, ร

6) เพดานแข็งแนวปุ่มเหงือก เสียงหยุดและเสียงเสียดแทรก ได้แก่ จ, ฉ, ช, ฌ

7) เพดานแข็ง เสียงก้อง ได้แก่ ย, ญ

8) เพดานอ่อน เสียงหยุด เสียงก้อง และ เส้นเสียงหรือสายเสียง ได้แก่ ก, ข, ฃ, ค, ฅ, ฆ, กร-, กล-, กว-, คร-, ขร-, คล-, ขล-, คว-, ง, อ, ห, ฮ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการแบ่ง Viseme ของเสียงสระจะแบ่งตามเสียงสระเดี่ยว เพราะว่าสระผสมและสระอักษรต่างเกิดขึ้นจากการออกเสียงผสมกันของสระเดี่ยว ดังนั้นจะได้ Viseme เสียงสระมี 9 กลุ่ม คือ

1) กลุ่มสระ –ะ ได้แก่ –ะ, -า, -ำ, -ไ, -ใ, เ-า

2) กลุ่มสระ –ิ ได้แก่ –ิ, -ี, เ-ียะ, เ-ีย

3) กลุ่มสระ –ึ ได้แก่ –ึ, -ื, เ-อะ, เ-อ

4) กลุ่มสระ –ุ ได้แก่ –ุ, -ู, -ัวะ, -ัว

5) กลุ่มสระ เ-ะ ได้แก่ เ-ะ, เ-

6) กลุ่มสระ แ-ะ ได้แก่ แ-ะ, แ-

7) กลุ่มสระ โ-ะ ได้แก่ โ-ะ, โ-

8) กลุ่มสระ เ-าะ ได้แก่ เ-าะ, -อ

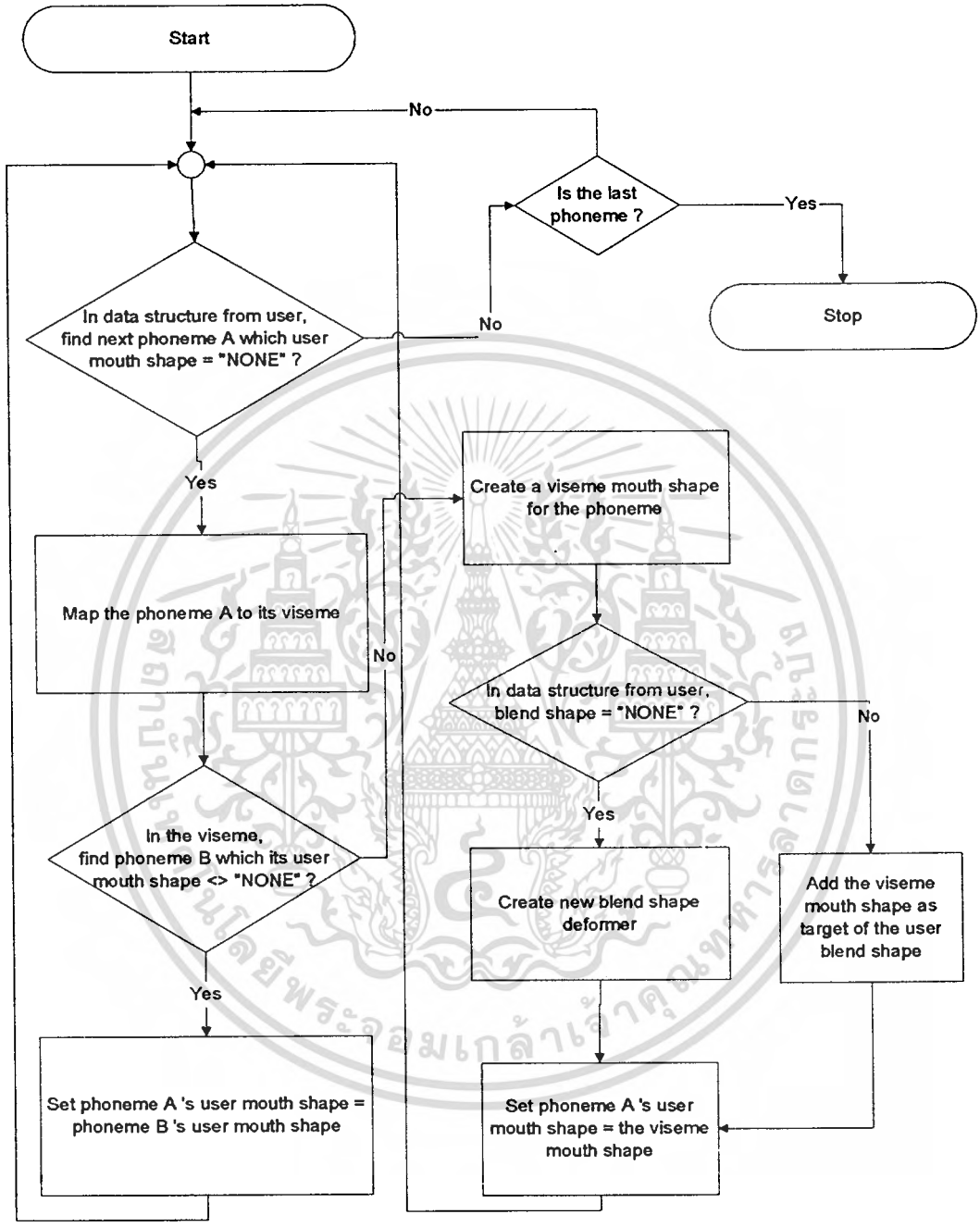
9) กลุ่มสระ เ-อะ ได้แก่ เ-อะ, เ-อ

สำหรับรูปการแบ่งกลุ่ม Viseme สามารถดูได้จากภาคผนวก ก

6.1.1 อัลกอริธึมการสร้างรูปปากสำหรับแต่ละหน่วยเสียงภาษาไทย

ขั้นตอนการทำงานของระบบย่อยการสร้างรูปปากคือ 1)ตรวจสอบโครงสร้างข้อมูล thaiPhonemes[phoneme_No][3] ที่ได้จากผู้ใช้ว่าโมเดลนี้มีการเตรียมทำภาพเคลื่อนไหวของปากหรือยัง โดยค้นหาว่ามีหน่วยเสียงใดบ้างที่มีค่ารูปปากเป็น "NONE" ถ้าพบจะนำไปจัดกลุ่มหาว่าหน่วยเสียงนี้อยู่ในกลุ่ม Viseme ใด จากนั้นจึงค้นหาหน่วยเสียงอื่นที่อยู่ในกลุ่ม Viseme นั้นว่ามีหน่วยเสียงใดบ้างที่มีค่ารูปปากไม่เป็น "NONE" ก็จะกำหนดให้ค่ารูปปากของหน่วยเสียงทั้งสองเหมือนกัน แล้วกลับไปทำข้อ 1 ใหม่ แต่ถ้าไม่พบจะสร้างรูปปาก Viseme ขึ้นมาใหม่ซึ่งตอบสนองต่อหน่วยเสียงนั้น แล้วตรวจสอบว่าโมเดลนี้มีการเตรียม Blend shape ให้อยู่แล้วหรือไม่ ถ้ามีอยู่แล้วให้เพิ่มรูปปาก Viseme นี้เข้าเป็นส่วน Target ของ Blend shape นั้น แต่ถ้าไม่มีให้สร้าง Blend shape ขึ้นมาใหม่ แล้วจึงกำหนดค่ารูปปากของหน่วยเสียงนั้นให้เป็นค่า Viseme ใหม่นี้ แล้วย้อนกลับไปทำข้อ 1 เพื่อหารูปปากสำหรับหน่วยเสียงต่อไปและทำเช่นนี้จนกระทั่งครบหมดทุกหน่วยเสียงเพื่อให้แน่ใจว่าทุกหน่วยเสียงมีรูปปากที่จะใช้ในการสร้างภาพเคลื่อนไหว ดังรูป 6.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1 Flow chart อัลกอริธึมโมดูลย่อยการสร้างรูปปาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.1 สัญลักษณ์หน่วยเสียงของพยัญชนะภาษาไทย

อักษรไทย	สัญลักษณ์หน่วยเสียง	
	ต้น	ตัวสะกด
ก	/k/	/k/
ข, ฃ, ค, ฅ, ฆ	/kh/	/k/
ง	/ng/	/ng/
จ	/c/	/t/
ฉ, ช, ฌ	/ch/	/t/
ซ, ศ, ษ, ส	/s/	/t/
ญ, ย	/j/	/j/
ฎ, ด	/d/	/t/
ฏ, ต	/t/	/t/
ฐ, ฑ, ฒ, ถ, ท, ธ	/th/	/t/
ณ, น	/n/	/n/
บ	/b/	/p/
ป	/p/	/p/
พ, ภ, ผ	/ph/	/p/
ฟ, ฝ	/f/	/p/
ม	/m/	/m/
ร	/r/	/n/
ล, ฬ	/l/	/n/
ว	/w/	/w/
ห, ฮ	/h/	-
อ	/z/	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.2 สัญลักษณ์หน่วยเสียงของพยัญชนะควบกล้ำภาษาไทย

อักษรไทย	สัญลักษณ์หน่วยเสียง	
	ต้น	ตัวสะกด
ปร-	/pr/	-
ปล-	/pl/	-
ตร-	/tr/	-
กร-	/kr/	-
กล-	/kl/	-
กว-	/kw/	-
พร-, ผร-	/phr/	-
พล-, ผล-	/phl/	-
ทร-	/thr/	-
คร-, ขร-	/khr/	-
คล-, ขล-	/khl/	-
คว-	/khw/	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.3 สัญลักษณ์หน่วยเสียงของสระภาษาไทย

สระเดี่ยว				สระคู่				สระอักษร	
เสียงสั้น		เสียงยาว		เสียงสั้น		เสียงยาว		เสียงสั้น	
-ะ	/a/	-า	/aa/	เ-ียะ	/ia/	เ-ีย	/iia/	ำ	/am/
ิ-	/i/	ี-	/ii/	เ-ือะ	/va/	เ-ือ	/vva/	ไ-, ใ-	/aj/
ึ-	/v/	ื-	/vv/	ั-วะ	/ua/	ั-ว	/uua/	เ-า	/aw/
ุ-	/u/	ู-	/uu/						
เ-ะ	/e/	เ-	/ee/						
แ-ะ	/x/	แ-	/xx/						
โ-ะ	/o/	โ-	/oo/						
เ-าะ	/@/	-อ	/@@/						
เ-อะ	/#/	เ-อ	/qq/						

6.2 การสร้างรูปปากสำหรับแต่ละเสียงพยางค์

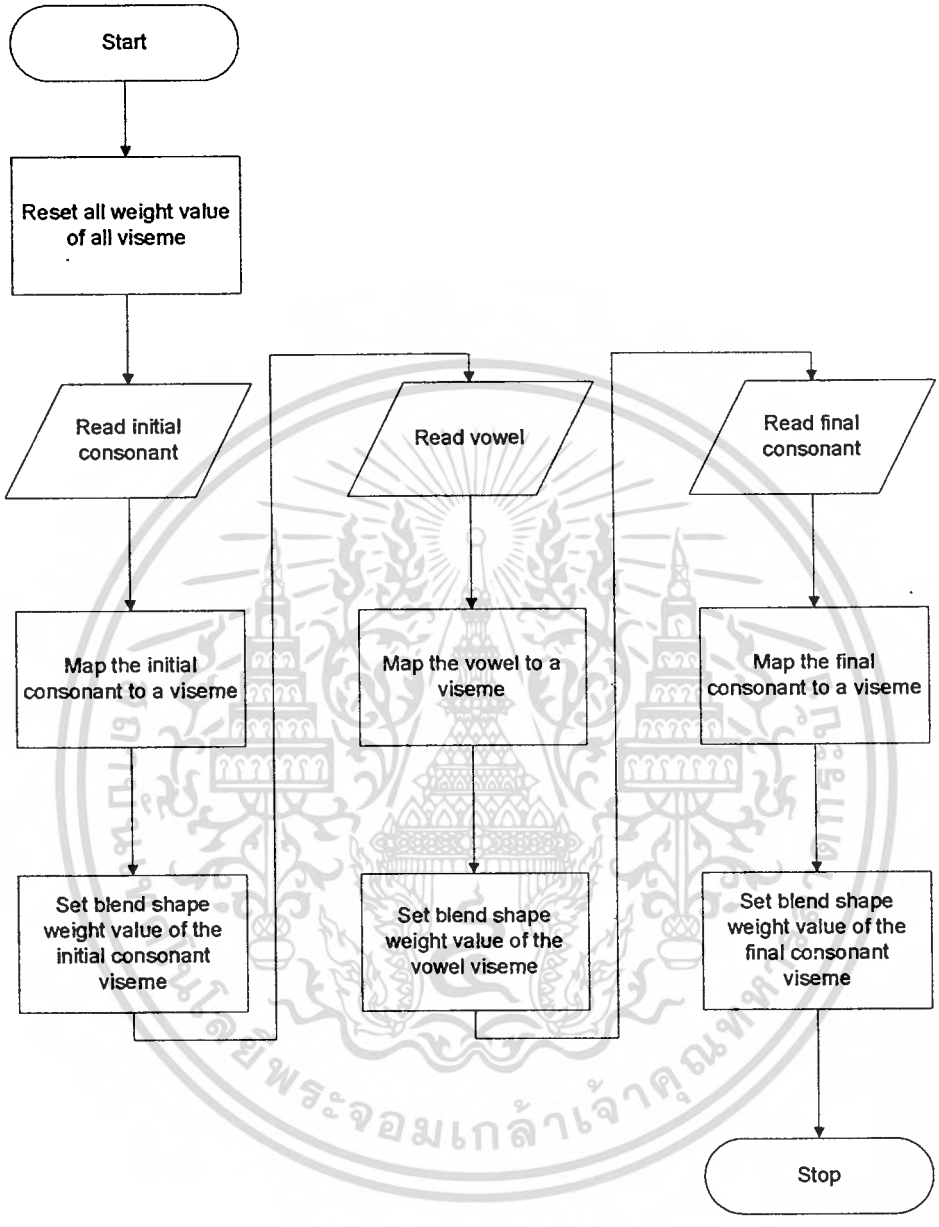
เสียงพยางค์เกิดจากการออกเสียงพยัญชนะ สระ ตัวสะกด และวรรณยุกต์ตามลำดับติดกันอย่างรวดเร็วจนกลายเป็นเสียงเดียวกัน ลักษณะรูปปากในการออกเสียงพยางค์จึงเป็นรูปปากเสียงพยัญชนะตามด้วยรูปปากเสียงสระและรูปปากเสียงตัวสะกด

6.2.1 อัลกอริธึมการสร้างรูปปากสำหรับแต่ละเสียงพยางค์

ขั้นตอนในการสร้างรูปปากเมื่อมี Blend shape ครบทุก Viseme แล้ว เริ่มด้วย

1) กำหนดค่า weight ของทุก Viseme ให้เป็น 0 เพื่อเริ่มต้นพยางค์ใหม่ จากนั้นจึงอ่านหน่วยเสียงพยัญชนะต้นแล้วนำค่าที่ได้มาจับคู่หา Viseme ของค่าหน่วยเสียงนั้นแล้วทำการกำหนดค่า weight ของ Viseme นี้ใหม่ให้เป็น 1 จากนั้นจึงอ่านหน่วยเสียงสระต่อไปแล้วนำค่าหน่วยเสียงสระนั้นไปจับคู่หา Viseme ของหน่วยเสียงสระนั้นแล้ว กำหนด Viseme นี้ให้มีค่า weight เป็น 1 จากนั้นจึงอ่านค่าตัวสะกดต่อไปแล้วจับคู่ค่าหน่วยเสียงตัวสะกดเพื่อหา Viseme แล้วทำการกำหนดค่า weight ของ Viseme นั้นให้เป็นค่า 1 แล้วจึงจบการผสมหน่วยเสียงให้ได้เป็นรูปปากเสียงพยางค์ ดังรูป 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 Flow chart อัลกอริธึมการสร้างรูปปากเสียงพยางค์

6.3 การสร้างส่วนควบคุมการผสมรูปปากด้วย Blend Shape Deformer

Blend shape deformer ใช้ Base shape และผสม Base shape นี้เข้ากับ Target shape อื่นๆ ตามค่า Weight ตัวของ Blend shape deformer จริงๆแล้ว คือ เครือข่ายเล็กๆ ของ Dependency nodes ใน Dependency Graph มี 3 ส่วนต่อ 1 Blend shape deformer คือมี Base shape, Target shape และค่า Weight ตัว Base shape จะเป็นส่วนซึ่งถูกปรับเปลี่ยนรูป ดังนั้นต้องมีอย่างน้อย 1 Base shape และ Base shape จะเปลี่ยนรูปขณะที่ Target shape และ พารามิเตอร์การเปลี่ยนรูปถูกปรับค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละ Base shape จะมีรายการของ Target shape ซึ่งกระทบกับการเปลี่ยนรูปของ Base shape นั้น แต่ละ Target shape จะมี 1 ค่า Weight ในการผสมให้ Base shape เมื่อค่า Weight นั้นเพิ่มขึ้น Target shape ก็จะมีอิทธิพลกับ Base shape มากขึ้น

6.4 โมดูลย่อยสร้างการเคลื่อนไหว

โมดูลย่อยนี้มีหน้าที่ 1) ค้นหาตำแหน่งที่เหมาะสมในการกำหนด Keyframe โดยที่ตำแหน่งนี้จะทำให้การเคลื่อนไหวของปากไปด้วยกันกับเสียงพูด 2) กำหนด Keyframe เพื่อสร้างการเคลื่อนไหว

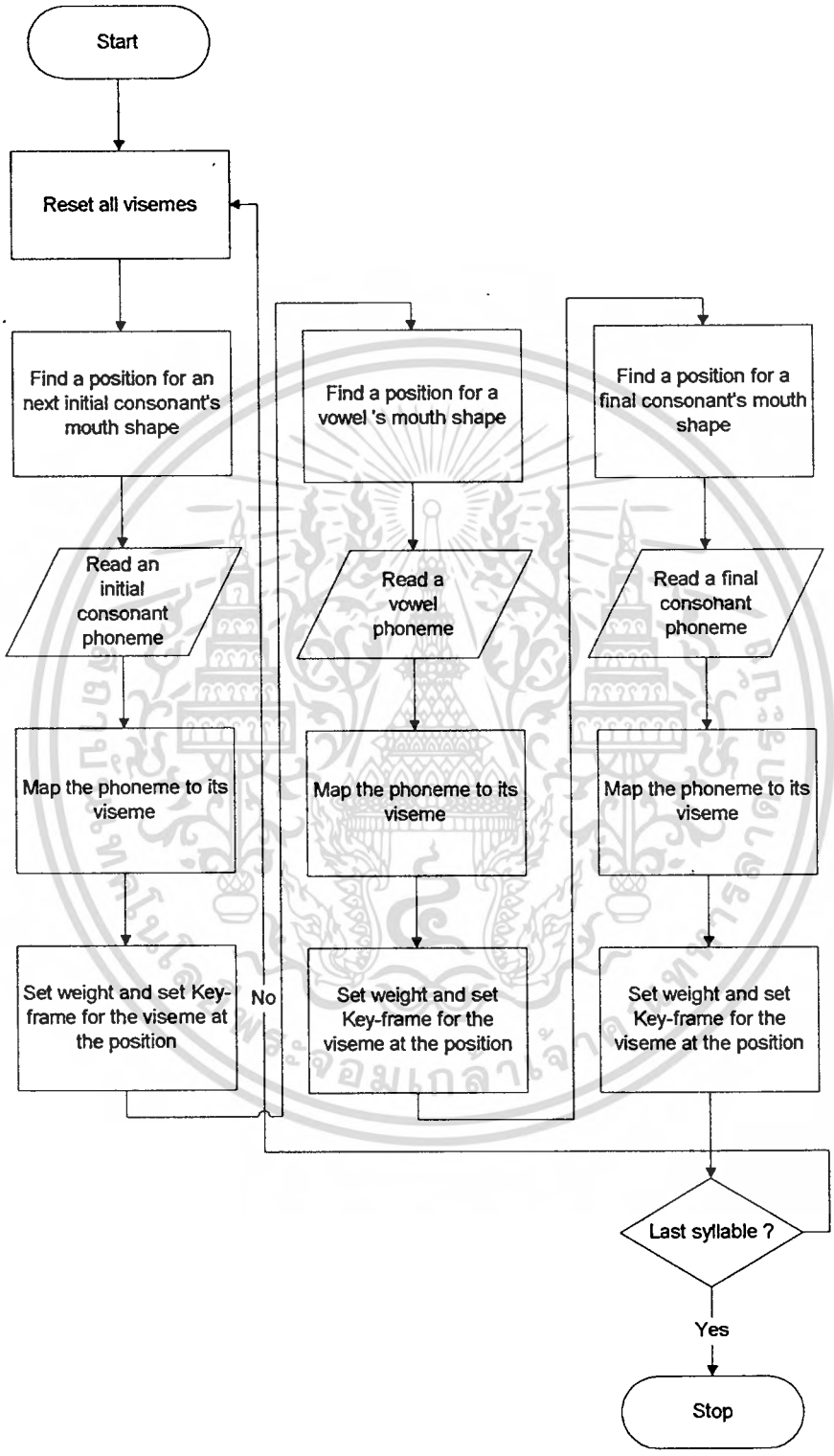
6.5 ขั้นตอนการทำงานของโมดูลย่อยสร้างการเคลื่อนไหว

ปัญหาของการสร้างภาพเคลื่อนไหวของปากในการพูดคือ จะทำอย่างไรให้ภาพที่ได้นั้นมีการขยับปากได้ตรงกับเสียงพูด สำหรับในโครงการนี้เมื่อมีการหาตำแหน่งเวลาที่เหมาะสมสำหรับรูปปากได้แล้วจะทำการกำหนดค่า Keyframe ไปด้วยกัน เพราะฉะนั้นทั้งสองหน้าที่ของโมดูลย่อยนี้จึงทำงานร่วมกัน

เริ่มจาก 1) กำหนดค่า Weight ของทุก Viseme ให้เป็น 0 เพื่ออยู่ในรูปปากปกติ จากนั้นหาตำแหน่งเวลาแรกสำหรับหน่วยเสียงพยัญชนะต้นแล้วอ่านค่าหน่วยเสียงพยัญชนะต้นของพยางค์แล้วจับคู่เข้ากับ Viseme รูปปาก เมื่อจับคู่ได้แล้วให้กำหนดค่า Weight ของ Viseme นั้นให้มากขึ้นแล้วจึงกำหนดตำแหน่งเวลานั้นให้เป็น Keyframe ต่อมาหาตำแหน่งเวลาสำหรับหน่วยเสียงสระแล้วอ่านค่าหน่วยเสียงสระของพยางค์แล้วจับคู่เข้ากับ Viseme รูปปาก เมื่อจับคู่ได้แล้วให้กำหนดค่า Weight ของ Viseme นั้นให้มากขึ้นแล้วจึงกำหนดตำแหน่งเวลานั้นให้เป็น Keyframe ต่อมาหาตำแหน่งเวลาสำหรับหน่วยเสียงตัวสะกดแล้วอ่านค่าหน่วยเสียงตัวสะกดของพยางค์แล้วจับคู่เข้ากับ Viseme รูปปาก เมื่อจับคู่ได้แล้วให้กำหนดค่า Weight ของ Viseme นั้นให้มากขึ้นแล้วจึงกำหนดตำแหน่งเวลานั้นให้เป็น Keyframe เมื่อครบทั้ง 3 ส่วนแล้วจึงตรวจสอบดูว่ายังมีพยางค์ให้กำหนดรูปปากอีกหรือไม่ ถ้ามีก็ย้อนกลับไปข้อ 1 ถ้าไม่มีก็จบการทำงาน ดังรูปที่ 6.3

สำหรับหน่วยเสียงวรรณยุกต์ในโครงการนี้จะไม่นำมาใช้เพราะว่าเป็นหน่วยเสียงที่เกิดจากตำแหน่งลิ้นที่อยู่ภายใน ซึ่งไม่กระทบกับรูปปากมากเมื่อมองด้วยตาเปล่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 Flow Chart อัลกอริธึมการค้นหาตำแหน่งและการกำหนด Keyframe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 สรุป

โมดูลย่อยการสร้างรูปปากจะเป็นโมดูลที่ใช้สำหรับสร้างรูปปากหน่วยเสียงโดยแบ่งตามลักษณะการออกเสียงภาษาไทย จากนั้นจึงสร้างรูปปากเสียงพยางค์โดยการผสมโมเดลรูปปากเหล่านั้นของผู้ใช้ให้มีความสามารถพร้อมสำหรับการทำภาพเคลื่อนไหวของปาก โดยใช้เทคนิคของ Blend shape deformer ในการปรับเปลี่ยนรูปร่างของปาก สำหรับรายละเอียดจะกล่าวถึงในบทที่ 7 ส่วนโมดูลย่อยสร้างการเคลื่อนไหวจะเป็นการกำหนด Keyframe เพื่อสร้างการเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การสร้างรูปปากมาตรฐานอัตโนมัติ

7.1 การสร้างรูปปากมาตรฐานอัตโนมัติ

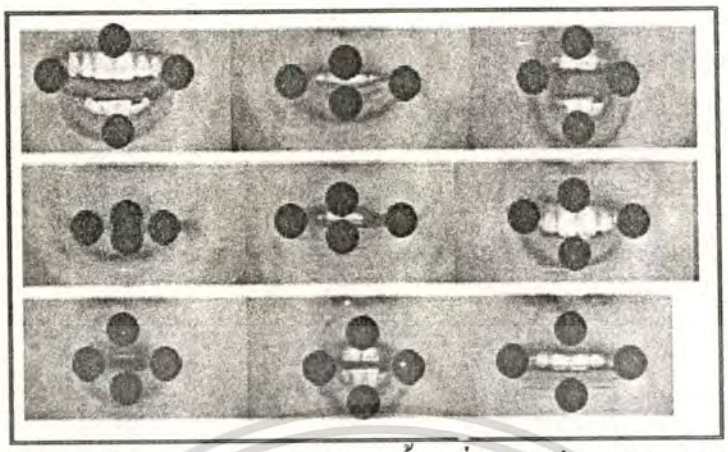
ในกรณีที่โมเดลธรรมดาที่ไม่มีการเตรียมรูปปากมาตรฐานสำหรับแต่ละหน่วยเสียงไว้ การที่จะสร้างภาพเคลื่อนไหว จึงจำเป็นต้องสร้างรูปปากมาตรฐานทั้ง 17 รูปขึ้นมาใหม่ เพื่อให้สามารถผสมเป็นรูปปากพยางค์และสร้าง Blend Shape ในโปรแกรมมายาได้ หลักการสร้างรูปปากอัตโนมัติในโครงการนี้ใช้วิธีกำหนด Control Vertex สำหรับตำแหน่งสำคัญของรูปปากโมเดลต้นแบบคือตำแหน่งมุมปากด้านซ้าย มุมปากด้านขวา กึ่งกลางริมฝีปากบน และกึ่งกลางริมฝีปากล่าง รวม 4 ตำแหน่ง เมื่อกำหนดตำแหน่งทั้ง 4 แล้วจึงทำการหมุน Control Vetices ทั้ง 4 เพื่อให้ได้รูปทรงของปากมาตรฐานทั้ง 17 รูป จากนั้นจึงสร้าง Blend Shape เพื่อใช้ในการขยับปาก ดังนั้นการสร้างรูปปากมาตรฐานอัตโนมัตินี้จึงเป็นขั้นตอนแรกในวิธีการทำให้ปากขยับ

7.2 หลักการขยับปาก

หลักการขยับรูปปากในโครงการนี้ใช้วิธี Blend Shape ซึ่งเป็นการสร้างรูปปากใหม่จากรูปปาก 2 รูปทรงขึ้นไป รูปปากใหม่นี้เกิดจากการผสมของรูปปากที่มีอยู่ การผสมจะมีน้ำหนักสัดส่วนของรูปปากที่มีอยู่ ถ้าผสมด้วยน้ำหนักของรูปปากใดมากก็จะทำให้รูปปากใหม่มีความใกล้เคียงรูปปากนั้น ดังนั้นการเพิ่มหรือลดน้ำหนักการผสมจะทำให้รูปปากใหม่มีการเปลี่ยนรูปทรงของปากต้นแบบไปสู่รูปทรงของปากเป้าหมายหรือเปลี่ยนรูปทรงกลับไปสู่รูปปากต้นแบบ การเปลี่ยนรูปทรงปากใหม่ทำอย่างต่อเนื่องทำให้ปากใหม่มีการเคลื่อนไหวขยับได้

อัลกอริธึมการหาตำแหน่งใหม่ของ Control Vertex รูปปากเป้าหมาย คิดจากการเคลื่อนย้ายตำแหน่งของ Control Vertex ของปากต้นแบบไปสู่ตำแหน่งใหม่ของรูปปากเป้าหมาย ทำโดยการเอาคิดค่า Vector x, y, z ของจุด Control Vertex ของรูปปากต้นแบบ 4 จุด ซึ่งแต่ละจุดแสดงถึงตำแหน่งของรูปปากดังรูปที่ 7.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



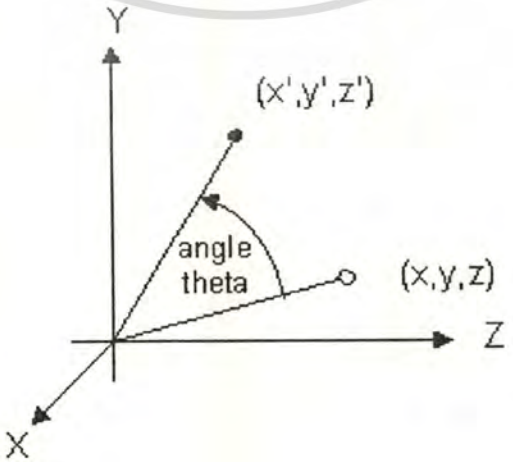
รูปที่ 7.1 ตำแหน่งของจุด Control Vertex ทั้ง 4 ที่สัมพันธ์กับรูปปากหน่วยเสียง

จากจุด Control Vertex ทั้ง 4 เมื่อนำมาหมุนรอบแกน X และแกน Y โดยจุดศูนย์กลางการหมุนอยู่ที่จุดกำเนิด (0, 0, 0) ด้วยค่ามุมที่แตกต่างกันจะสามารถทำให้ได้รูปปากหลายแบบตามหน่วยเสียง โดยแต่ละจุดมีสมการของการหมุนดังนี้

สมการหมุนจุด Control Vertex (x,y,z) รอบแกน X ทวนเข็มนาฬิกาเป็นมุม θ ให้ได้ตำแหน่งจุดใหม่ (x',y',z') คือ

$$X' = X$$
$$Y' = Z\sin(\theta) + Y\cos(\theta)$$
$$Z' = Z\cos(\theta) - Y\sin(\theta)$$

สมการนี้แสดงได้ดังรูปที่ 7.2



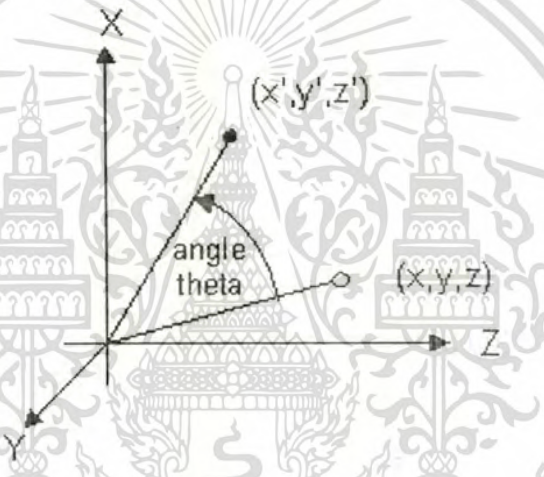
รูปที่ 7.2 ลักษณะการหมุนจุดรอบแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสมการหมุนจุด Control Vertex (x,y,z) รอบแกน Y ทวนเข็มนาฬิกาเป็นมุม θ ให้ได้ตำแหน่งจุดใหม่ (x',y',z') คือ

$$X' = Z \sin(\theta) + X \cos(\theta)$$
$$Y' = Y$$
$$Z' = Z \cos(\theta) - X \sin(\theta)$$

สมการนี้แสดงได้ดังรูปที่ 7.3



รูปที่ 7.3 ลักษณะการหมุนจุดรอบแกน Y

7.3 วิธีการกำหนดข้อมูลเสียงสำหรับรูปปาก

การกำหนดว่าเสียงใดจะใช้รูปปากใด ทำโดยหลักการวิเคราะห์หน่วยเสียงของพยัญชนะต้น สระ และตัวสะกด แล้วนำมาผสมกันด้วยวิธี Blend Shape ในสัดส่วนค่าน้ำหนักการผสมที่แตกต่างกันเพื่อให้ได้รูปปากที่มีรูปทรงเหมือนกับธรรมชาติของเสียงพูดจริงมากที่สุด ซึ่งในโครงการนี้จะผสมด้วยสัดส่วนน้ำหนักค่าพยัญชนะต้นเท่ากับ 0.9 สระเท่ากับ 0.4 และตัวสะกดเท่ากับ 0.9 ซึ่งเป็นการกำหนดแบบคงที่ทุกเสียงพยางค์จะผสมด้วยค่าน้ำหนักนี้

7.4 ข้อมูลการกำหนดมุมและรูปปากใหม่ที่ได้

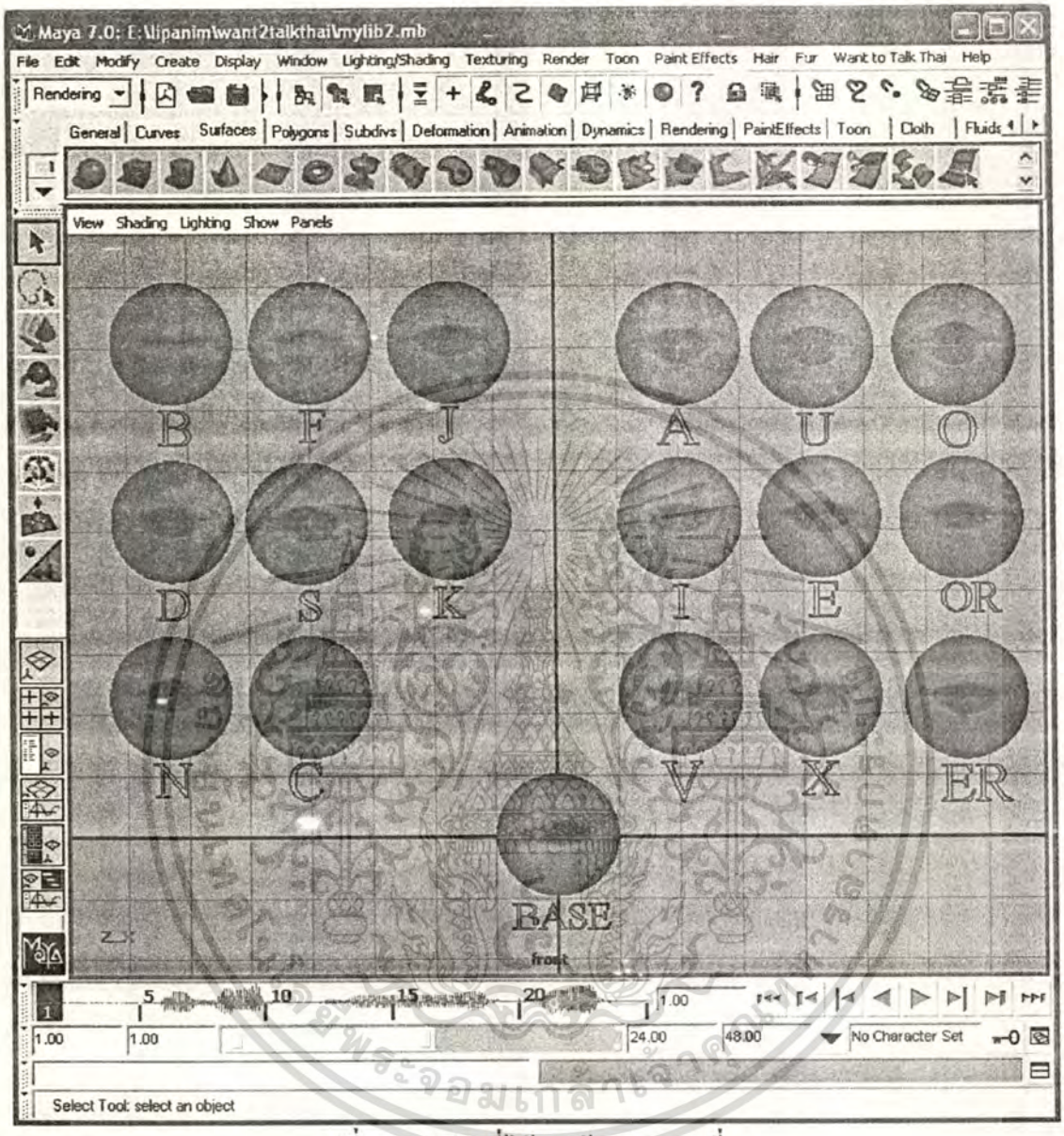
ข้อมูลของมุมการหมุนของ Control Vertices ทั้ง 4 ในโครงการนี้แสดงดังตารางที่ 7.1 ตัวเลขของมุมมีหน่วยเป็นองศา และผลลัพธ์รูปปากที่ได้จากการกำหนดค่ามุมเหล่านี้แสดงในรูปที่ 7.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 7.1 ข้อมูลการกำหนดค่ามุมรูปปาก

Mouth Shape	Angle of Control Vertex 1	Angle of Control Vertex 2	Angle of Control Vertex 3	Angle of Control Vertex 4
BASE	0	0	0	0
B	0	0	-20	20
F	0	0	0	30
J	5	-5	10	-10
D	0	0	5	-15
S	0	0	10	0
K	0	0	0	10
N	10	-10	15	-10
C	11	-11	16	-11
A	5	-5	5	-30
U	25	-25	25	-25
O	30	-30	30	-30
I	0	0	5	-10
E	-10	10	5	-20
OR	10	-10	5	-5
V	0	0	0	-10
X	0	0	0	-15
ER	0	0	0	-20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 รูปปากที่ได้จากข้อมูลตารางที่ 7.1

7.5 การสร้างไฟล์เสียง

คุณลักษณะอย่างหนึ่งของโปรแกรมเสริมมายาสำหรับการสร้างภาพเคลื่อนไหวของปากในการพูดนี้คือ สามารถแปลงข้อความให้กลายเป็นไฟล์เสียงพูดได้ ซึ่งความสามารถนี้เป็นของไลบรารี Vaja ของโปรแกรมวาจาซึ่งได้พัฒนาโดย NECTEC ซึ่งหลักการสร้างไฟล์เสียงคือจะบันทึกความถี่เสียงของหน่วยเสียงต่างๆเก็บเป็นฐานข้อมูลคล้าย Dictionary และนำความถี่หน่วยเสียงเหล่านั้นมาผสมกันเป็นพยางค์ตามการวิเคราะห์ประโยค แล้วจากนั้นจึงบันทึกเป็นไฟล์ .wav สำหรับรายละเอียดการสร้างไฟล์เสียงสามารถศึกษาเพิ่มเติมได้จากเว็บไซต์ของ NECTEC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6 หลักการควบคุมการเคลื่อนไหวสำหรับโมเดล 3 มิติ

โปรแกรมเสริมจะไม่มีประโยชน์เลยหากใช้งานได้เฉพาะโมเดล ดังนั้นการที่จะทำให้โปรแกรมเสริมสามารถทำงานได้กับทุกโมเดลคือโปรแกรมเสริมต้องทำงานอยู่ในระดับที่สามารถเข้าถึงองค์ประกอบพื้นฐานของทุกโมเดลได้ องค์ประกอบพื้นฐานนี้คือ Control Vertex ซึ่งมีอยู่ทุกโมเดล 3 มิติในโปรแกรมมายา

โครงการนี้ได้พัฒนาโปรแกรมเสริมให้ทำงานอยู่ในระดับ Control Vertex ซึ่งทำให้แน่ใจได้ว่าสามารถใช้ได้กับทุกโมเดลแบบ NURBS ซึ่งเป็นโมเดลที่เหมาะสำหรับการสร้างรูปมนุษย์ โปรแกรมเสริมนี้จะควบคุมการเคลื่อนที่ของ Control Vertex ให้เป็นไปตามรูปแบบปากของหน่วยเสียงที่กำหนดไว้ เมื่อ Control Vertex เคลื่อนที่จะทำให้พื้นผิวที่อยู่ในการควบคุมมีการปรับตัวไปตามการเคลื่อนที่ของ Control Vertex นั้นด้วย

สำหรับโมเดลที่ผู้ใช้ต้องการคุณภาพรูปปากที่ดี จะต้องสร้างรูปปากของแต่ละหน่วยเสียงเองและสร้าง Blend Shape Deformer สำหรับรูปปากเหล่านั้น จากนั้นจึงใช้โปรแกรมเสริมของโครงการนี้ช่วยกำหนด Keyframe เพื่อทำการสร้างภาพเคลื่อนไหวได้เช่นกัน ซึ่งคุณภาพของรูปปากจะได้ตามที่ผู้ใช้ต้องการ

7.7 สรุป

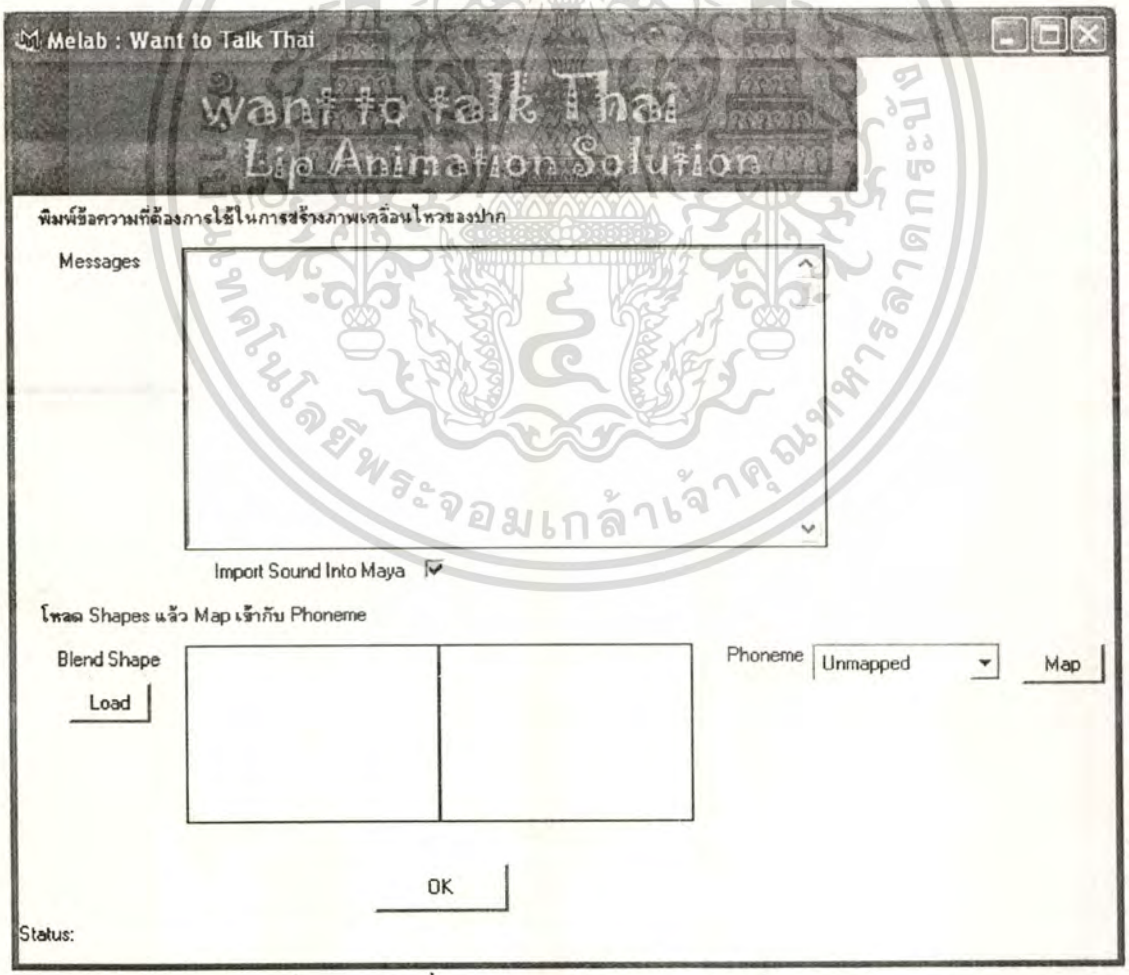
ปัญหาของการสร้างภาพเคลื่อนไหวของปากในการพูดคือ เมื่อโมเดลไม่มีการเตรียมรูปปากมาตรฐานของแต่ละหน่วยเสียงเอาไว้ให้นั้น จะทำอย่างไรให้ได้รูปปากของหน่วยเสียงที่ต้องการ ในการแก้ปัญหาสำหรับโครงการนี้คือ 1) ใช้การควบคุมการเคลื่อนที่ของ Control Vertex 4 จุดในแนวแกน X, Y เพื่อให้ได้รูปปากหน่วยเสียงที่ต้องการเหล่านั้น จากนั้นทำการสร้าง Blendshape เพื่อให้โมเดลต้นแบบสามารถขยับปากพูดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
ผลการศึกษาและพัฒนาโครงการ

8.1 โปรแกรมสร้างภาพการเคลื่อนไหวของปากในการพูดภาษาไทย

โปรแกรมสร้างภาพการเคลื่อนไหวของปากในการพูดภาษาไทยนี้เป็นโปรแกรมเสริมสำหรับโปรแกรมมายา (Plug-in for Maya) เวอร์ชั่น 7 เป็นโปรแกรมที่ช่วยให้กระบวนการสร้างภาพเคลื่อนไหวของปากทำได้ง่ายและรวดเร็วขึ้น โดยเราสามารถนำโปรแกรมนี้ไปใช้กับตัวแบบ (Model) ที่มีอยู่แล้วได้ทันที โปรแกรมเสริมที่ได้จากโครงการนี้จะมีลักษณะดังรูปที่ 8.1



รูปที่ 8.1 หน้าต่างโปรแกรมเสริม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

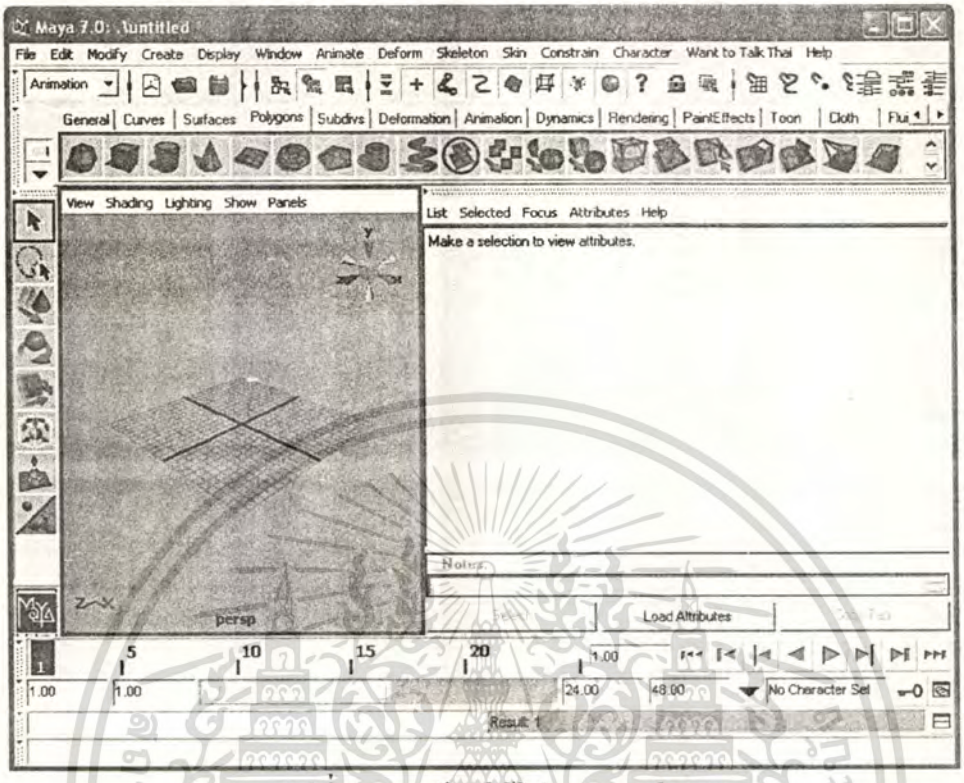
8.2 ขั้นตอนการใช้งาน

1) เมื่อเปิดโปรแกรมมายาแล้ว ให้โหลดโปรแกรมเสริมโดยเลือกที่เมนู Window->Settings/Preferences->Plug-in Manager จะปรากฏหน้าต่าง Plug-in Manager ดังรูป

รูปที่ 8.2 หน้าต่าง Plug-in Manager

2) ให้เราคลิ้กปุ่ม Browse จากนั้นเลือกโฟลเดอร์ที่เก็บไฟล์โปรแกรมเสริมชื่อ want2talkthai.mll แล้วคลิ้กปุ่ม Load แล้วปิดหน้าต่าง Plug-in Manager ถ้าโหลดสำเร็จจะได้เมนูใหม่ของโปรแกรมเสริมชื่อ Want to Talk Thai ดังรูป 8.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.3 เมนูเรียกใช้โปรแกรมเสริม

3) เลือกเมนู Want to Talk Thai->Lip Animation Options จะปรากฏหน้าต่างของโปรแกรมเสริมขึ้นมา ดังรูป 8.1

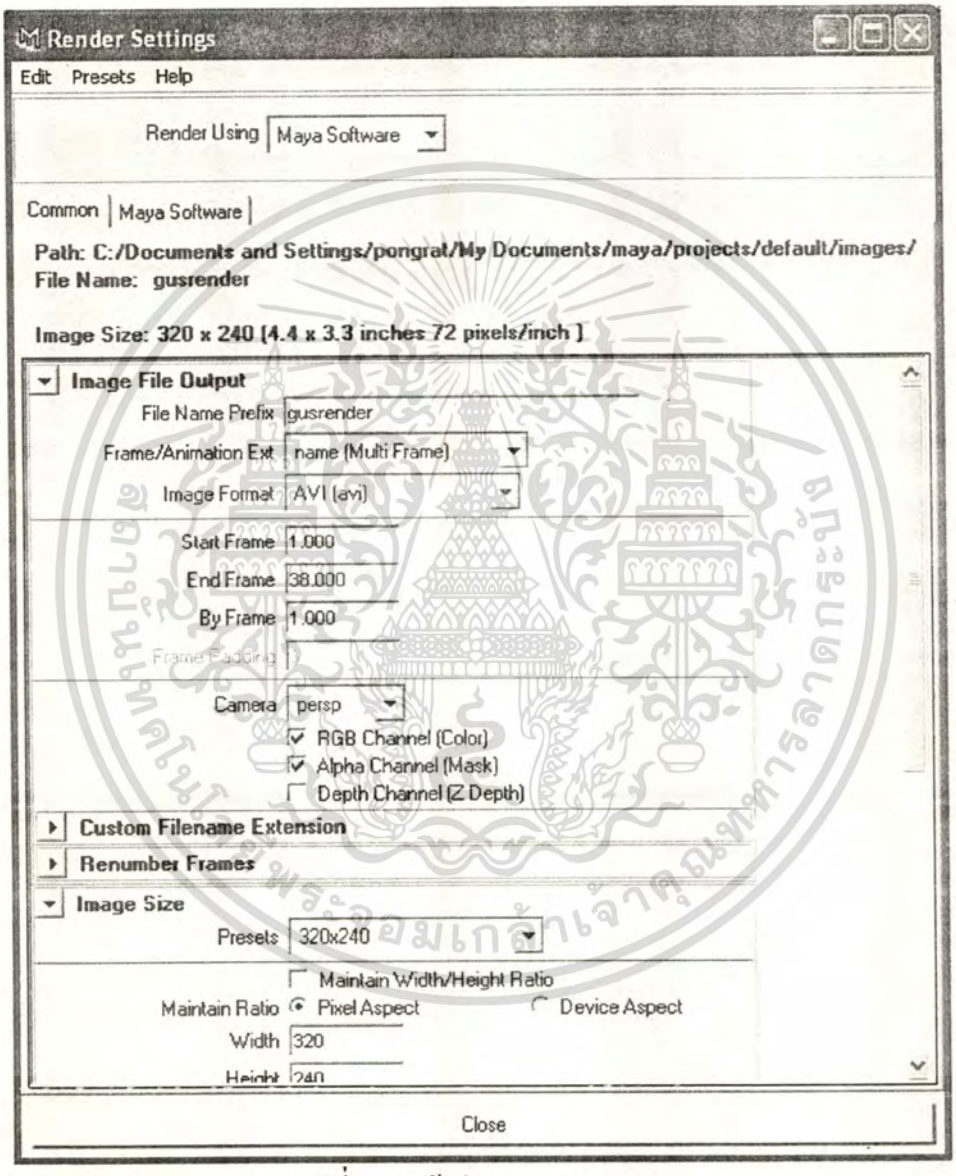
4) การใส่ค่าต่างๆในโปรแกรมเสริมมีรายละเอียดดังนี้

- ในกล่องข้อความ Message ให้พิมพ์ข้อความที่ต้องการให้โมเดลนี้พูด
- Check Import sound into Maya ถ้าต้องการให้มีการ โหลดเสียงพูดสังเคราะห์เข้าไปใน Time line
- ในส่วน Blend shape คลิ้กปุ่ม Load เพื่อให้แสดงรายการ Blend shape ทั้งหมดที่มีใน Scene ขณะนั้น
- เลือก Blend shape อีกช่องจะแสดงรายการทุกพื้นผิวที่มีใน Scene ขณะนั้น ให้เลือกพื้นผิวที่ได้เตรียมทำรูปปากของหน่วยเสียงเอาไว้แล้ว จากนั้นเลือก Phoneme ให้ตรงกับรูปปากหน่วยเสียง แล้วจึงคลิ้กปุ่ม Map
- ทำการ Map รูปปากหน่วยเสียงกับ Phoneme จนครบแล้วคลิ้กปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 การสร้างภาพการเคลื่อนไหว

1) ตั้งค่าการสร้างภาพเคลื่อนไหวโดยเลือกเมนู Window->Rendering editors->Render Settings จะมีหน้าต่างของ Render Setting ดังรูป 8.4



รูปที่ 8.4 หน้าต่าง Render Setting

2) ในหัวข้อ Common ใส่ค่าในส่วนของ Image File Output และ Image Size เพื่อเป็นการกำหนดรายละเอียดในการสร้างภาพเคลื่อนไหว โดยค่า Image Format จะเลือกเป็น AVI(avi)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เลือกหัวข้อ Maya Software แล้วเลือกค่าในส่วนของ Anti-aliasing Quality เป็น Production Quality เพื่อให้คุณภาพของภาพไฟล์เคลื่อนไหวที่ดีที่สุด แล้วปิดหน้าต่าง Render Setting

4) เลือกเมนู Render->Batch Render เพื่อทำการเริ่มสร้างภาพการเคลื่อนไหว

5) ไฟล์ภาพเคลื่อนไหวที่ได้จะมีนามสกุล .avi อยู่ในโฟลเดอร์ My Documents\maya\projects\default\images

รูปที่ 8.5 ภาพที่ได้จากการพูดคำว่า "สวัสดี"

8.4 สรุป

โปรแกรมเสริมของโครงการนี้ วิธีใช้งานก็เป็นดังที่กล่าวมาแล้ว ผู้ใช้ต้องใส่ข้อมูลรูปปากให้ถูกต้องเพราะ ไม่เช่นนั้นโปรแกรมจะเข้าใจผิดในเรื่องของการจับคู่รูปปากกับหน่วยเสียงในการทำภาพเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
บทสรุป

9.1 สรุป

การพัฒนาโครงการระบบโปรแกรมเสริมนี้ เป็นการศึกษาและพัฒนาในด้านคอมพิวเตอร์กราฟิกส์ โดยเกี่ยวข้องกับการสร้างโปรแกรมเสริมสำหรับซอฟต์แวร์สร้างภาพเคลื่อนไหว 3 มิติ Maya ซึ่งเป็นโปรแกรมที่นิยมใช้กันมากในด้านการทำภาพเคลื่อนไหว โดยพัฒนาด้วย Maya API, MEL script, และ Vaja API ส่วน Compiler ที่ใช้คือ MS Visual Studio .NET 2003 Visual C++ และพัฒนาบนระบบปฏิบัติการ Windows XP โดยโปรแกรมเสริมนี้ช่วยในการสร้างภาพการเคลื่อนไหวของปากในการพูดภาษาไทย ทำให้กระบวนการสร้างภาพเคลื่อนไหวนี้ทำได้รวดเร็วขึ้น

9.2 ข้อจำกัดและปัญหาของโครงการ

ข้อจำกัดของการพัฒนาโครงการระบบโปรแกรมเสริมนี้ คือ 1) เรื่องของเวลา ซึ่งก็ต้องเข้าใจว่าชีวิตมนุษย์แปรผกผันกับเวลา ดังนั้นโครงการนี้เมื่อใช้เวลาพอสมควรแล้วก็คงต้องยุติถึงแม้ว่าโครงการจะไม่สำเร็จดีเยี่ยมตามต้องการ 2) เรื่องทักษะในการเขียนโปรแกรม การออกแบบระบบและการทำเอกสาร ซึ่งผู้พัฒนาก็พยายามพัฒนาอยู่ 3) เรื่องทฤษฎีซึ่งสำหรับโครงการนี้ก็พยายามหาที่อ้างอิงบ้าง ศึกษาด้วยตนเองบ้าง ดังนั้นโครงการนี้จึงเป็นในลักษณะ Pilot Project มากกว่าเป็น Project ธรรมดา

สำหรับปัญหาของโครงการนี้ก็คือ 1) เรื่องเครื่องมือที่ใช้ในการออกแบบในลักษณะ Object-Oriented Programming ซึ่งจะใช้เพียง CRC และ Collaboration Graphs เท่านั้น ทั้งนี้เพราะโครงการนี้เป็นโครงการด้านการเขียนโปรแกรม ซึ่งการออกแบบจะเน้นที่อัลกอริธึมในการทำงานมากกว่าปกติ และการอธิบายอัลกอริธึมนั้นใช้การเขียน Flow Chart ช่วย 2) เรื่องการ Implement ของ Application ซึ่งมีลักษณะเป็นโปรแกรมเสริมมายา ซึ่งต้องเข้าใจกลไกลของโปรแกรมมายาเป็นอย่างดีเพื่อให้โปรแกรมเสริมที่พัฒนาขึ้นสามารถทำงานได้ร่วมกับฟังก์ชันปกติของโปรแกรมมายาเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.3 ข้อเสนอแนะ

สำหรับโครงการพัฒนาระบบงานนี้ยังมีส่วนที่น่าจะพัฒนาได้มากกว่านี้ เช่น ความเป็นธรรมชาติในการพูด, ความสวยงามของโมเดลที่สร้างใหม่ ซึ่งต้องใช้อัลกอริธึมที่ซับซ้อนกว่าในการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

นิตยา กาญจนะวรรณ และ Eynon, Matthew J.. 2548. **เรียนภาษาไทย**. พิมพ์ครั้งที่ 2. กรุงเทพฯ: โอเดียนสโตร์.

Alias Systems. 2004. **Learning Maya 6 Character Rigging and Animation**. Canada:Maya Press.

Alias Systems. 2004. **Maya API White Paper**. USA.

Alias|Wavefront. 2000. **The art of Maya**. USA.

Alias|Wavefront. 2000. **Using Maya : MEL**. USA.

Angel, Edward. 2002. **Interactive Computer Graphics**. 3rd ed. USA:Addison Wesley.

Bahrami, Ali. 1999. **Object Oriented Systems Development**. Singapore:Irwin/McGraw-Hill.

Douglas, Young A.. 1995. **Object Oriented Programming with C++ and OSF Motif**. 2nd ed. New Jersey:Prentice-Hall.

Kundert-Gibbs J. and Lee P.. 2000. **Mastering Maya 3**. USA:Sybex.

Parke, Frederic I. and Keith Waters. 1996. **Computer Facial Animation**. USA:A K Peters.

Pradit Mittrapiyanurak et al. 2000. "Issues in Thai Text-to-Speech Synthesis:TheNECTEC Approach." **NECTEC Technical Journal**. 11(7): 36-47.

Weiss, Mark Allen. 1999. **Data Structure and Algorithm Analysis in C++**. USA:Addison Wesley Longman.

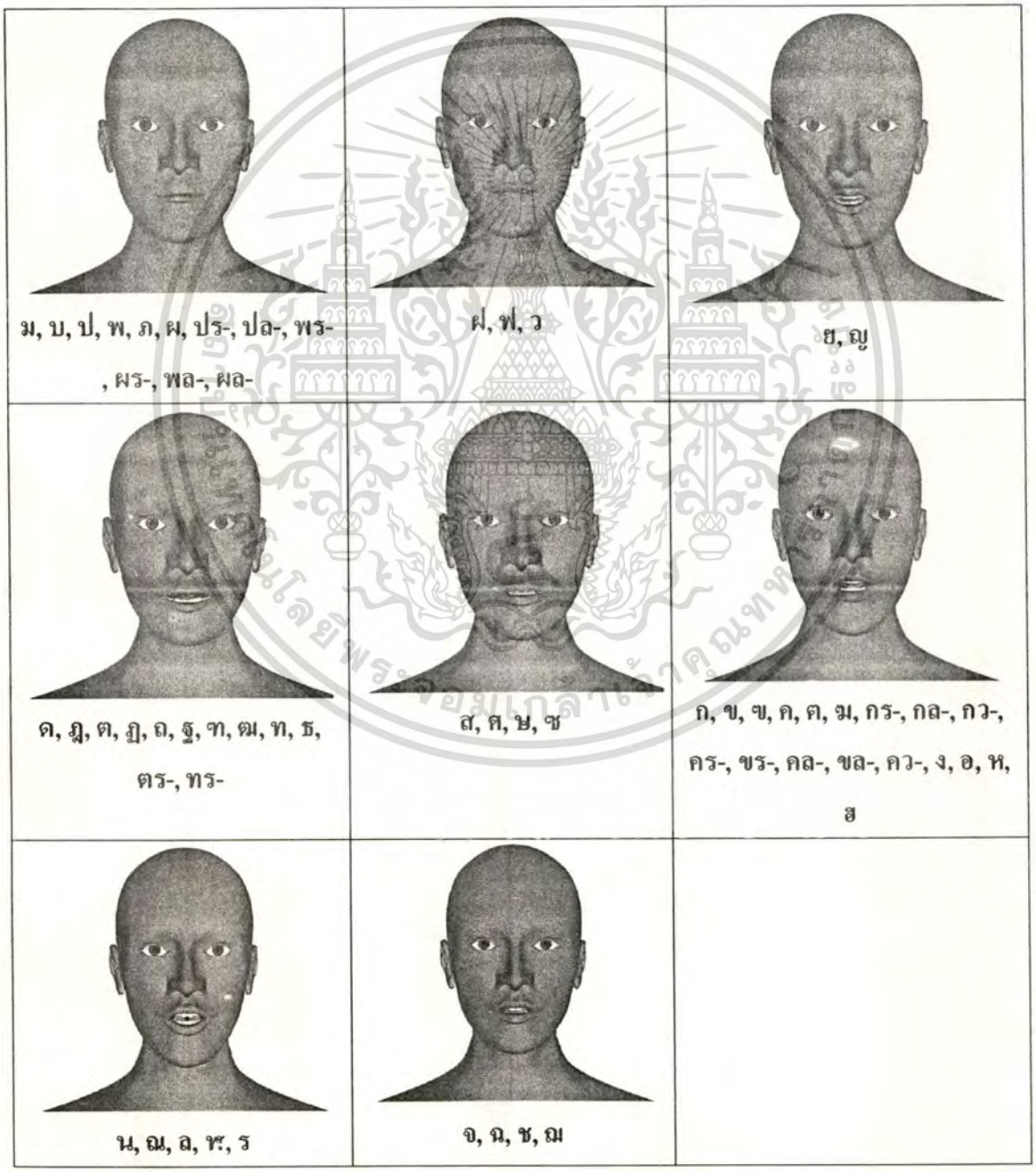
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

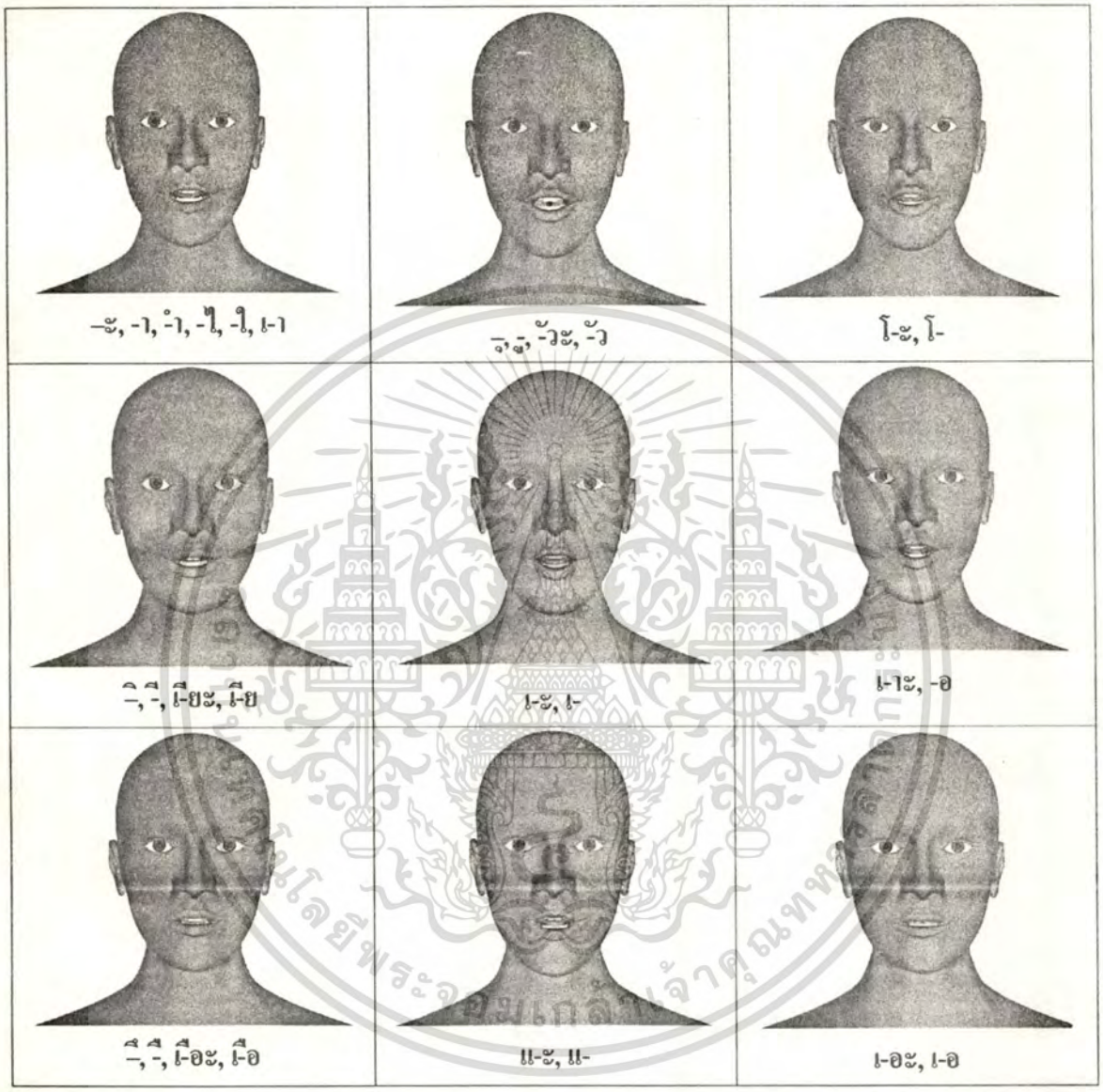
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การแบ่งกลุ่ม Viseme เสียงในภาษาไทย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

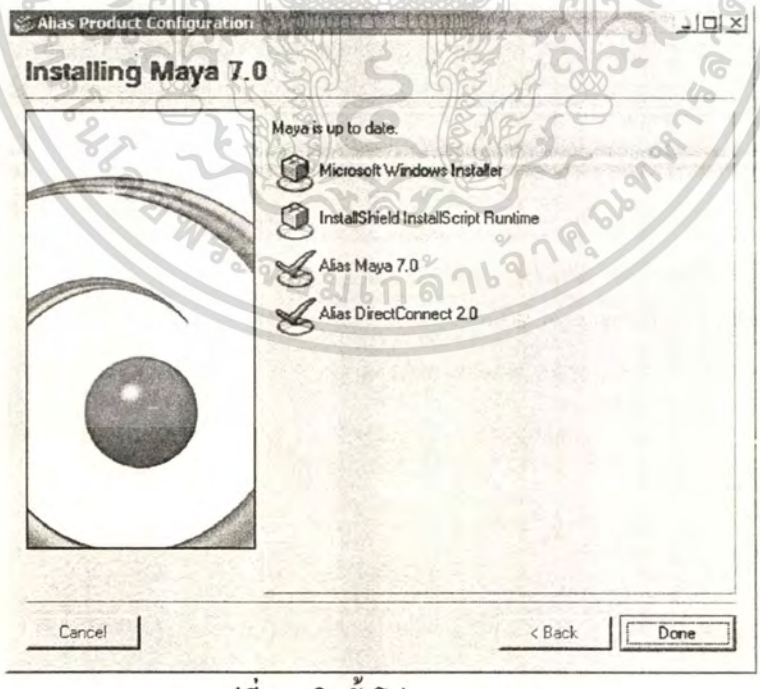
ภาคผนวก ข
คู่มือการติดตั้ง

สิ่งที่จำเป็นในการติดตั้งใช้งานโปรแกรมเสริมของโครงการนี้มี 3 โปรแกรมคือ

- โปรแกรม Maya เวอร์ชัน 7.0
- โปรแกรม Vaja Thai TTS เวอร์ชัน 3.1
- โปรแกรมเสริม want2talkthai

1. การติดตั้งโปรแกรม Maya เวอร์ชัน 7.0

ไฟล์สำหรับการติดตั้งคือ myr_maya70_win.exe ให้ทำการติดตั้งจนใช้งานได้ตามปกติ ดังรูป



รูปที่ การติดตั้งโปรแกรม Maya

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การติดตั้งโปรแกรม Vaja Thai TTS เวอร์ชัน 3.1

ดาวน์โหลดไฟล์สำหรับการติดตั้งคือ vaja.exe จากเว็บไซต์ http://vaja.nectec.or.th แล้วให้ทำการติดตั้งจนใช้งานได้ตามปกติ ดังรูป

รูปที่ การติดตั้งโปรแกรม Vaja

3. การติดตั้งโปรแกรมเสริม

โปรแกรมเสริมอยู่ในไฟล์ w2tt.zip เมื่อ extract จะได้โฟลเดอร์ want2talkthai ซึ่งภายในประกอบด้วยโฟลเดอร์ others และ plug-in จากนั้นให้ทำดังนี้

1) Copy ไฟล์ทั้งหมดในโฟลเดอร์ others ไปใส่ในโฟลเดอร์ C:\Program Files\Alias\Maya7.0\scripts\others

2) โปรแกรมเสริมจะอยู่ในโฟลเดอร์ want2talkthai\plug-in ชื่อไฟล์ want2talkthai.mll

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นาย พงศ์รัฐ เมืองเจริญ
วันเดือนปีเกิด	31 มกราคม 2517
สถานที่เกิด	นครศรีธรรมราช
วุฒิการศึกษาระดับปริญญาตรี	ศิลปศาสตรบัณฑิต(ปรัชญาและศาสนา)
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยเกษตรศาสตร์
ปีที่สำเร็จการศึกษา	2537

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้