

โครงการทางวิศวกรรม

เรื่อง

ทัมบ์สไวป์: คีย์บอร์ดภาษาไทยอัจฉริยะใช้งานง่ายสำหรับแอนดรอยด์

ThumbSwipe: Intuitive and Intelligent Thai Input Method for Android Devices

โดย

นายเขมินท์ คงจำเนียร 5230641921

นายฐิติคมน์ ญาณสมบัติ 5231013921

นายอรรถพงศ์ ลิ้มศุภนาค 5231061021

อาจารย์ที่ปรึกษาโครงการ ผศ.ดร.สุกรี สิ้นธุภิญโญ ลายมือชื่อ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชาโครงการวิศวกรรมคอมพิวเตอร์

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

ข้อเสนอโครงการวิจัย

ชื่อไทย: ทัมบ์สไวป์: คีย์บอร์ดภาษาไทยอัจฉริยะใช้งานง่ายสำหรับแอนดรอยด์

ชื่ออังกฤษ: ThumbSwipe: Intuitive and Intelligent Thai Input Method for Android Devices

ปัญหาและความสำคัญของปัญหา

ในปัจจุบันนี้ สมาร์ทโฟน (smartphone) เข้ามามีบทบาทสำคัญต่อชีวิตประจำวันของผู้คนมากขึ้น ด้วยความสามารถที่ใกล้เคียงกับคอมพิวเตอร์พกพาในรูปของโทรศัพท์เคลื่อนที่ ไม่ว่าจะเป็น การรับส่งอีเมล การติดตามข่าวสารจากทางเว็บไซต์ต่างๆ รวมถึงความสามารถของแอปพลิเคชัน (application) ต่างๆ ที่ติดตั้งอยู่บนตัวสมาร์ทโฟนเอง ที่กล่าวนำข้างต้นเป็นปัจจัยหลักที่ผลักดันให้สมาร์ทโฟนกลายมาเป็นส่วนหนึ่งในชีวิตประจำวันของคนในปัจจุบัน

โดยสมาร์ทโฟนที่วางขายตามท้องตลาดนั้นมีอยู่หลากหลายระบบปฏิบัติการ หนึ่งในนั้นคือ แอนดรอยด์ (Android) ซึ่งจัดเป็นหนึ่งในระบบปฏิบัติการยอดนิยมของสมาร์ทโฟน โดยครองส่วนแบ่งตลาดอยู่ถึง 68.1% ของตลาดสมาร์ทโฟนโลกเมื่อช่วงไตรมาสที่สองของปี 2555 [4] คำถามก็คือ ในปัจจุบันนี้มีวิธีการป้อนข้อมูลเข้าภาษาไทยที่ดีพอหรือยังสำหรับแอนดรอยด์ เนื่องจากวิธีการป้อนข้อมูลเข้าภาษาไทยสำหรับแอนดรอยด์ในปัจจุบันนี้เป็นแป้นพิมพ์ซอฟต์แวร์ (soft-keyboard) ซึ่งยังคงยึดตามรูปแบบของแป้นพิมพ์เกษมณี ทำให้เกิดปัญหาหลักๆ อยู่สามประการ คือ

ประการแรก ปัญหาขนาดของแป้นพิมพ์ เนื่องจากแป้นพิมพ์นั้นถูกออกแบบตามแป้นพิมพ์ของคอมพิวเตอร์ที่มีขนาดใหญ่ แต่เมื่อถูกนำมาทำเป็นแป้นพิมพ์ซอฟต์แวร์ซึ่งต้องแสดงบนหน้าจอที่มีขนาดเล็กนั้น ทำให้เกิดปัญหามาตาม คือ ตัวแป้นที่มีขนาดเล็กเมื่อเทียบกับขนาดนิ้วโป้งหรือนิ้วชี้ที่ใหญ่กว่าแป้นตัวอักษรมาก จนทำให้ไม่สามารถมองเห็นแป้นตัวอักษรได้ชัดเจน และมีโอกาสที่จะกดผิดพลาดไปกดแป้นตัวอักษรข้างเคียงได้ ถึงแม้จะมีการพยายามนำเทคโนโลยีต่างๆ มาใช้ในการช่วยเพิ่มความถูกต้อง เช่น การเดาคำ หรือ การปรับขนาดแป้นพิมพ์อัตโนมัติ

ประการที่สอง ตำแหน่งของแป้นตัวอักษร เนื่องจากปัจจุบันรูปแบบที่นิยมเป็นรูปแบบแป้นพิมพ์เกษมณี ทำให้ผู้ต้องใช้เวลามากในการจำตำแหน่งอักษรตัวต่างๆ บนแป้นพิมพ์ จึงจะพิมพ์ข้อความได้อย่างคล่องแคล่ว ถึงแม้จะมีผู้ใช้งานจำนวนถึงที่สามารถใช้งานแป้นพิมพ์แบบนี้ได้ทันทีเพราะคุ้นชินกับรูปแบบแป้นพิมพ์แบบเดียวกันจากคอมพิวเตอร์แล้ว อย่างไรก็ตามยังมีผู้ใช้จำนวนมากที่ไม่สามารถใช้งานได้อย่างคล่องแคล่วเนื่องจากไม่มีประสบการณ์การใช้งานแป้นพิมพ์รูปแบบดังกล่าว ผู้ใช้เหล่านี้มักเป็นผู้ที่ไม่ได้ใช้คอมพิวเตอร์เป็นประจำ หรือไม่เคยใช้เลย ใช้แป้นพิมพ์ครั้งแรกบนสมาร์ทโฟน

ประการสุดท้าย แป้นพิมพ์ทั่วไปถูกออกแบบให้ใช้กับนิ้ว 10 นิ้ว การใช้แป้นพิมพ์แบบนี้กับหน้าจอโทรศัพท์ที่มีรูปแบบการใช้ต่างกัน คือ ใช้นิ้วโป้ง หรือนิ้วชี้ในการป้อนข้อมูล จึงไม่เหมาะสมเท่าที่ควร

จากทั้งสามประการที่ได้กล่าวมาแล้ว จะเห็นว่ายังคงมีความยากลำบากในการป้อนข้อมูลให้กับแอนดรอยด์ด้วยวิธีที่มีอยู่ในปัจจุบัน

ดังนั้น จึงเกิดเป็นแนวคิดที่ว่า หากสามารถออกแบบ รวมถึงพัฒนาวิธีการป้อนข้อมูลเข้าภาษาไทยให้กับแอนดรอยด์ขึ้นมาใหม่ โดยวิธีดังกล่าวนี้ ผู้ใช้จะต้องสามารถใช้งานได้อย่างคล่องแคล่วโดยเสียเวลาเรียนรู้ไม่มากนัก โดยรูปแบบการป้อนข้อมูลเข้าดังกล่าวจะต้องเอื้อต่อการใช้งานมือเดียว (ทั้งนี้ เพื่อความสะดวกสบาย และความคล่องตัวในการใช้งาน) ได้แล้วละก็ ย่อมจะเป็นการอำนวยความสะดวกให้กับการใช้สมาร์ทโฟนในชีวิตประจำวันของสังคมผู้ใช้แอนดรอยด์ไทยได้อย่างแน่นอน

ทฤษฎีที่เกี่ยวข้อง

1. การแบ่งคำไทย [2]

ในการทำให้คอมพิวเตอร์ที่รู้จักแต่เพียงรหัสอักษร แต่ไม่เข้าใจถึงความหมายของคำ ให้สามารถแบ่งคำไทยได้นั้น สามารถแบ่งออกได้เป็นสามแนวทางหลักๆ คือ

แนวทางที่หนึ่ง การใช้กฎ โดยการสร้างพยางค์ไทยอันประกอบไปด้วยพยัญชนะต้น สระ วรรณยุกต์ ตัวสะกด การันต์ โดยแนวทางนี้ทำได้ง่ายที่สุด ทำงานได้เร็วที่สุด แต่สามารถแบ่งได้เฉพาะคำพยางค์เดียวเท่านั้น ไม่สามารถจัดการกับคำหลายพยางค์ได้ รวมถึงยังไม่สามารถจัดการกับความกำกวมของพยัญชนะในคำที่สามารถเป็นได้ทั้งพยัญชนะต้น และตัวสะกด ภายในคำได้ เช่น ตัวอักษร ‘ก’ ภายในคำว่า ‘ตากลม’ ที่สามารถแบ่งคำออกเป็น ‘ตาก-ลม’ หรือ ‘ตา-กลม’ ก็ได้

แนวทางที่สอง การใช้พจนานุกรม (dictionary) โดยจะต้องเตรียมรายการของคำเอาไว้ เมื่อจะทำการแบ่งคำก็จะทำการเปรียบเทียบข้อความที่ต้องการแบ่งกับรายการของคำที่มีอยู่ในพจนานุกรม โดยแนวทางนี้จะสามารถจัดการกับคำหลายพยางค์ได้ อย่างไรก็ตาม แนวทางดังกล่าวจะยังไม่สามารถจัดการกับปัญหาความกำกวมได้

แนวทางที่สาม การใช้เทคนิคการเรียนรู้ของเครื่อง (machine learning) เป็นแนวทางที่นิยมใช้กันมากในปัจจุบัน โดยการใช้คลังข้อความขนาดใหญ่ที่มีการแบ่งคำ ไว้อย่างถูกต้องแล้ว เพื่อให้เครื่องได้เรียนรู้ด้วยตัวเอง จากการเก็บสถิติและการคำนวณค่า ความน่าจะเป็นของการปรากฏร่วมของคำที่อยู่ติดๆ กัน โดยประสิทธิภาพของวิธีนี้จะขึ้นกับความถูกต้องและขนาดของคลังข้อความ รวมถึงอัลกอริธึมในการคำนวณค่าทางสถิติ

2. การออกแบบส่วนต่อประสานกับผู้ใช้ และ ประสบการณ์การใช้งาน (User Interface and User Experience design)

ปัจจัยหลักหนึ่งซึ่งทำให้ผู้ใช้ป้อนข้อมูลเข้าในอุปกรณ์พกพาได้อย่างมีประสิทธิภาพ คือส่วนต่อประสานกับผู้ใช้ ซึ่งจำเป็นที่จะต้องได้รับการออกแบบให้ใช้ง่าย รวดเร็ว ผู้ใช้สามารถใช้งานได้อย่างคล่องแคล่วแม้ว่าไม่เคยมีประสบการณ์การใช้งานมาก่อน การออกแบบส่วนต่อประสานกับผู้ใช้ได้แก่ การกำหนดตำแหน่ง ลักษณะ และพฤติกรรมของส่วนต่างๆ บนหน้าจออุปกรณ์ซึ่งมีปฏิสัมพันธ์กับผู้ใช้ ยกตัวอย่างเช่น แป้นพิมพ์อักษร “ก” ต้องมีการกำหนดตำแหน่งของแป้นบนหน้าจอ กำหนดขนาด และสีของแป้น ระยะห่างของแป้นกับแป้นอักษรอื่น ต้องมีการกำหนดตำแหน่ง ขนาด และสีของอักษร “ก” บนแป้น กำหนดพฤติกรรมของแป้นเมื่อถูกกด อาจเปลี่ยนขนาดหรือสีของแป้นชั่วคราว พฤติกรรมของแป้นเมื่อระบบคาดเดาความน่าจะเป็นที่ผู้ใช้จะกดแป้นนี้เป็นแป้นถัดไป อาจเน้นให้แป้นมีความเด่นชัดมากกว่าแป้นอื่นๆ

การออกแบบส่วนต่างๆ ตามที่ได้กล่าวไว้ข้างต้นนั้น ส่งผลโดยตรงกับการใช้งานของผู้ใช้ โดยส่งผลให้ผู้ใช้มีประสบการณ์การใช้งานที่แตกต่างกันออกไป การออกแบบส่วนต่อประสานที่ดีอาจส่งผลให้ผู้ใช้ได้รับประสบการณ์ที่ดีขึ้นในทางตรงกันข้าม การออกแบบส่วนต่อประสานที่ไม่ดีย่อมส่งผลให้ผู้ใช้ได้รับประสบการณ์ที่ไม่ดี อย่างไรก็ตาม การทดสอบว่าการออกแบบส่วนต่อประสานแบบหนึ่งๆ นั้นเป็นการออกแบบที่ดีหรือไม่นั้น ไม่สามารถบ่งบอกได้อย่างชัดเจนเนื่องจากผู้ใช้แต่ละคนมีพฤติกรรมการใช้งานที่แตกต่างกัน การที่จะทดสอบว่าการออกแบบแบบใดเป็นการออกแบบที่ดีกว่าจึงต้องอาศัยการสำรวจ โดยวิธีหนึ่งที่ให้ผลลัพธ์ที่ดีและแม่นยำตรงคือการให้ผู้ใช้ทดลองใช้ และศึกษาจากพฤติกรรมการใช้ การศึกษาจากความเห็นของผู้ใช้ในแบบสำรวจเพียงอย่างเดียวอาจให้ผลลัพธ์ที่ไม่ตรงความเป็นจริงนัก เนื่องจากการออกแบบส่วนต่อประสานใหม่นี้เป็นอาจได้ผลลัพธ์เป็นส่วนต่อประสานใหม่ ซึ่งไม่เคยมีมาก่อน ความเห็นของผู้ใช้ซึ่งใช้การเปรียบเทียบกับประสบการณ์การใช้งานในอดีต จึงอาจไม่ได้ผลตามที่ต้องการเท่าใดนัก

ถึงแม้ส่วนต่อประสานจะมีความสำคัญต่อประสบการณ์การใช้งานมาก แต่ปัจจัยอื่นๆ ก็ส่งผลต่อประสบการณ์การใช้งานเช่นกัน ยกตัวอย่างเช่น ประสิทธิภาพการทำงานของโปรแกรมซึ่งเป็นส่วนที่ทำนายเนื่องจากอุปกรณ์แอนดรอยด์มีตั้งแต่เครื่องที่ฮาร์ดแวร์มีประสิทธิภาพสูงไปจนต่ำ อีกปัจจัยหนึ่งได้แก่ความสามารถในการคาดเดาอักษรหรือคำถัดไปในประโยคซึ่งผู้ใช้อาจพิมพ์ ซึ่งสามารถเพิ่มประสบการณ์การใช้งานในรูปแบบที่วัดผลได้ซึ่งจะกล่าวต่อไป

3. ปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง (Artificial Intelligence and Machine Learning)

การทำนายคำศัพท์หรืออักษรต่อไปที่ผู้ใช้อาจพิมพ์อาศัยปัญญาประดิษฐ์ โดยใช้ร่วมกับการเรียนรู้ของเครื่องเพื่อให้ผลลัพธ์ที่ตรงกับผู้ใช้แต่ละคนมากที่สุด และมีความแม่นยำเพิ่มขึ้นเมื่อใช้ไป การทำนายเช่นนี้อาจใช้หลักการที่เรียกว่า n-gram

n-gram เป็นแบบจำลองการทำนายทางภาษาโดยอาศัยความน่าจะเป็น (probabilistic language model) โดยหมายถึงจำนวนสิ่งของที่ติดกัน n สิ่ง จากประโยคหรือคำพูดที่ให้มา โดยสิ่งของในความหมายนี้จะสื่อถึง ตัวอักษร คำ วลี ซึ่ง n-gram มักจะรวบรวมมาจาก text และ speech corpus n-gram มักจะถูกนำมาใช้ในการทำนายสิ่งของถัดไป (n-1)

4. การยศาสตร์ (Ergonomics)

คือ การศึกษาเกี่ยวกับการประสานกัน หรืออันตรกิริยาระหว่างมนุษย์และเครื่องมืออุปกรณ์ภายใต้สิ่งแวดล้อมที่มนุษย์นั้นทำงานอยู่ โดยครอบคลุมถึงการออกแบบผลิตภัณฑ์ให้สอดคล้องกับสรีระ การเคลื่อนไหวของมนุษย์ เพื่อให้ผู้ใช้สามารถใช้งานผลิตภัณฑ์ดังกล่าวได้อย่างมีประสิทธิภาพสูงสุด

งานวิจัยที่เกี่ยวข้อง

1. Method for designing an ergonomic one-finger keyboard and apparatus therefor [7]

สำหรับการจัดวางแป้นพิมพ์คอมพิวเตอร์ในปัจจุบัน ทั้งแบบ Qwerty และแบบ Dvorak นั้นต่างก็ถูกออกแบบมาสำหรับการพิมพ์ด้วยนิ้วมือทั้งสิบนิ้ว แต่สำหรับแป้นพิมพ์บนอุปกรณ์พกพา ที่ใช้การพิมพ์ด้วยนิ้วเหมือน หรืออุปกรณ์เสริม เช่น ปากกา stylus เป็นหลักนั้น การนำแป้นพิมพ์แบบ Qwerty หรือ Dvorak มาใช้นั้น จะทำให้ผู้ใช้ต้องลากนิ้วหรือปากกา stylus ไปทั่วบริเวณของแป้นพิมพ์เป็นแน่ จึงเกิดเป็นแนวคิดที่จะลดการลากนิ้วที่ไม่จำเป็นขึ้น โดยจะทำการจัดเรียงแป้นตัวอักษรใหม่ ให้ตัวอักษรที่อยู่ใกล้กันนั้นเป็นตัวอักษรที่มีโอกาสที่จะถูกพิมพ์ต่อกันนั้นสูง รวมถึงจัดวางตำแหน่งของแป้นพิมพ์ให้อยู่ในระยะที่นิ้วมือของผู้ใช้สามารถเอื้อมถึง

2. Using paper mockups for evaluating soft keyboard layouts [6]

ในงานวิจัยชิ้นนี้ได้มีการนำแบบร่างบนกระดาษมาใช้วัดผลความเร็วในการพิมพ์บนแป้นพิมพ์แบบ Qwerty และแป้นพิมพ์แบบ Opti เพื่อให้สำหรับการพัฒนาแป้นพิมพ์บนเครื่อง PDA โดยแป้นพิมพ์แบบ Qwerty และแป้นพิมพ์แบบ Opti มีลักษณะ ดังภาพ

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	
Z	X	C	V	B	N	M			
space									

Q	F	U	M	C	K	Z
space	O	T	H	space		
B	S	R	E	A	W	X
space	I	N	D	space		
J	P	V	G	L	Y	

ภาพแสดงการจัดวางแป้นพิมพ์แบบ Qwerty (ซ้าย) และ Opti (ขวา)

โดยในการทดลองวัดผลนั้นได้ใช้ผู้ทดลองทั้งสิ้น 12 คนทำการจับคู่กัน โดยคนหนึ่งทำหน้าที่จับเวลา ส่วนอีกคนหนึ่งทำหน้าที่เป็นคนทดลองใช้แป้นพิมพ์ โดยการใช้ปากกาหรือดินสอ (แทน stylus ของ PDA) ไล่พิมพ์ไปตามแป้นตัวอักษรแต่ละแป้น ตามวลีหรือประโยคต่างๆ ที่สุ่มมาใช้ทดลอง แล้วจึงนำจำนวนคำในวลี/ประโยค และเวลาที่ใช้ในการพิมพ์มาคำนวณหาค่าความเร็วในหน่วยคำต่อนาที

โดยผลที่ได้จากการทดลองนั้นเป็นไปในทางเดียวกับการวัดผลแป้นพิมพ์ด้วยวิธีอื่นๆ ซึ่งมีความเป็นกิจจะลักษณะมากกว่าการทดสอบด้วยกระดาษ จึงยอมรับได้ว่าการวัดผลการจัดวางแป้นพิมพ์โดยใช้แบบร่างบนกระดาษนั้นเป็นวิธีที่มีประสิทธิภาพและสามารถทำได้รวดเร็ว

3. Design and evaluation of Devanagari virtual keyboards for touch screen mobile phones [3]

งานวิจัยชิ้นนี้เป็นการพัฒนาการจัดวางแป้นพิมพ์เพื่อรองรับภาษาฮินดี โดยมีเป้าหมายในการออกแบบคือเพื่อให้ได้มาซึ่งแป้นพิมพ์ที่ใช้งานง่าย และไม่จำเป็นต้องมีความรู้ในการใช้งานมาก่อน โดยความท้าทายในการออกแบบแป้นพิมพ์สำหรับภาษาฮินดีนั้น จะอยู่ที่ลักษณะเฉพาะของตัวภาษาเอง ที่มีความซับซ้อนสูง และมีจำนวนตัวอักษรในภาษาอยู่เป็นจำนวนมาก ทำให้การจัดเรียงแป้นตัวอักษรนั้นจะต้องใช้ค่าความถี่ในการถูกใช้งานเข้ามาช่วยในการพิจารณาการจัดวางตำแหน่งด้วย

4. Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop [8]

งานวิจัยชิ้นนี้จะศึกษาถึงการเพิ่มประสิทธิภาพสำหรับจากจัดวางแป้นพิมพ์รูปแบบต่างๆ โดยจะพิจารณาถึงรูปแบบการป้อนข้อมูลขาเข้าที่อาศัยการลากนิ้ว เปรียบเทียบกับการป้อนข้อมูลขาเข้าด้วยการแตะแป้นพิมพ์เสมือน โดยได้ทำการเปรียบเทียบความเร็วในการป้อนข้อมูลด้วยวิธีต่างๆ ที่เหมาะกับการใช้งานด้วยมือเดียว อันได้แก่ การแตะแป้นการลากนิ้วมือข้างซ้าย และการลากนิ้วมือข้างขวา ซึ่งได้ผลออกมา ดังตาราง

โดยจากตาราง จะเห็นว่าการป้อนข้อมูลขาเข้าด้วยการลากนิ้วบนแป้นพิมพ์แบบ Qwerty นั้นรวดเร็วกว่าการป้อนข้อมูลขาเข้าด้วยการแตะแป้นบนแป้นพิมพ์แบบเดียวกันถึง 17% ซึ่งเร็วขึ้นมากอย่างเห็นได้ชัดจนคุ้มค่าที่ผู้ใช้จะเปลี่ยนมาใช้วิธีการป้อนข้อมูลขาเข้าด้วยการลากนิ้ว ในขณะที่การป้อนข้อมูลขาเข้าด้วยลากนิ้วบนแป้นพิมพ์ Opt || นั้นเร็วกว่าการป้อนข้อมูลขาเข้าด้วยการแตะแป้นบนแป้นพิมพ์แบบ Qwerty อยู่ถึง 50% ซึ่งมากพอที่ผู้ใช้จะยอมเปลี่ยนรูปแบบการจัดวางแป้นพิมพ์ รวมถึงรูปแบบวิธีการป้อนข้อมูลขาเข้าที่มีประสิทธิภาพมากกว่า

Keyboard Layout	Key Distance / #Letters - 1	Tapping Speed	Stroking Speed (left-handed)	Stroking Speed (right-handed)	Stroking Speed / Tapping Speed
Qwerty	320%	34.7 wpm	40.6 wpm	40.7 wpm	117.3%
Dvorak	425%	33.0 wpm	36.3 wpm	36.2 wpm	109.7%
Wide Alphabetic	344%	34.3 wpm	39.8 wpm	39.5 wpm	115.4%
Montgomery	238%	35.6 wpm	46.0 wpm	46.0 wpm	129.4%
Getschow et al.	187%	37.8 wpm	51.0 wpm	50.8 wpm	134.5%
Chubon	195%	39.2 wpm	50.4 wpm	50.2 wpm	127.9%
Fitaly	192%	41.9 wpm	50.4 wpm	50.4 wpm	120.3%
Cirrin	284%	35.4 wpm	43.0 wpm	43.0 wpm	121.6%
Quikwriting	310%	n/a	28.3 wpm	28.3 wpm	n/a
OPTI I	185%	41.9 wpm	51.3 wpm	51.3 wpm	122.5%
OPTI II	175%	41.6 wpm	52.6 wpm	52.7 wpm	126.5%
Lewis et al.	197%	37.6 wpm	50.0 wpm	50.0 wpm	133.1%
Square Alphabetic	259%	35.8 wpm	44.1 wpm	44.0 wpm	122.9%
Metropolis I	229%	38.6 wpm	46.9 wpm	46.5 wpm	120.6%
Hooke	236%	38.1 wpm	45.9 wpm	46.2 wpm	121.3%
Metropolis II	221%	39.2 wpm	47.4 wpm	47.5 wpm	121.1%
ATOMIK	234%	38.5 wpm	45.8 wpm	46.2 wpm	119.8%
GAG I	204%	40.1 wpm	49.0 wpm	49.2 wpm	122.6%
GAG II	187%	43.0 wpm	51.5 wpm	51.3 wpm	119.2%
Square ATOMIK	190%	36.0 wpm	50.5 wpm	50.6 wpm	140.8%
Hexagon Qwerty	334%	32.4 wpm	39.3 wpm	39.4 wpm	121.5%
Quasi-Qwerty	249%	37.0 wpm	45.6 wpm	45.6 wpm	123.3%
Square OSK	173%	40.9 wpm	52.8 wpm	53.0 wpm	129.6%
Hexagon OSK	174%	37.6 wpm	52.3 wpm	52.6 wpm	139.2%

ตารางเปรียบเทียบความเร็วในการป้อนข้อมูลฯเข้าด้วยวิธีต่างๆ บนแป้นพิมพ์แต่ละแบบ

5. ชุดพัฒนาซอฟต์แวร์สำหรับแอนดรอยด์ (Android Software Development Kit หรือ Android SDK)

ชุดพัฒนาซอฟต์แวร์สำหรับแอนดรอยด์ คือชุดคำสั่งที่พัฒนาขึ้นมาด้วยภาษา Java สำหรับการพัฒนาซอฟต์แวร์ซึ่งทาง Google เป็นผู้เผยแพร่ให้นักพัฒนาสามารถเรียกใช้ได้ฟรี โดยภายในชุดพัฒนาฯ นั้นจะมีโปรแกรมจำลองการทำงาน (emulator) รวมอยู่ด้วยด้วย ทำให้ในระหว่างการพัฒนาซอฟต์แวร์สามารถจำลองการทำงานของระบบปฏิบัติการแอนดรอยด์ และทดลองใช้งานซอฟต์แวร์ที่ได้ทำการพัฒนาขึ้นโดยไม่จำเป็นต้องมีสมาร์ตโฟนจริง

6. ATOMIK [9]

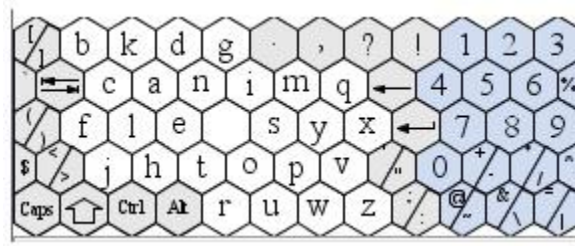
เนื่องจากแป้นพิมพ์แบบ Qwerty และ Dvorak นั้นถูกออกแบบมาให้พิมพ์ด้วยมือทั้งสองข้างสลับกันไปมา เมื่อนำแป้นพิมพ์ดังกล่าวมาใช้กับอุปกรณ์พกพาจึงเกิดปัญหาที่ผู้ใช้ (ซึ่งใช้มือข้างเดียว หรือใช้ปากกา stylus เป็นหลักในการสัมผัสหน้าจอ) จะต้องเลื่อนมือไปทางซ้ายที่ขวาที่ระหว่างการพิมพ์

ATOMIK (Alphabetically Tuned and Optimized Mobile Interface Keyboard) จึงถูกออกแบบมาเพื่อแก้ปัญหาดังกล่าว โดยคำนึงถึงหลักสำคัญสามประการ ดังนี้

ประการแรก ใช้การเคลื่อนไหวของมือได้คุ้มค่า โดยการจัดวางแป้นตัวอักษรนั้นวางให้ตัวอักษรต่างๆ นั้นอยู่ในบริเวณเดียวกัน ดังภาพ ทำให้การใช้ปากกา stylus ไล่แตะแป้นแต่ละแป้นนั้นจะอาศัยการเคลื่อนที่ของมือที่น้อยลง

ประการที่สอง การจัดวางจะเรียงตามตัวอักษร โดยตัวอักษรจาก A ถึง Z จะไล่ไปตั้งแต่มุมบนซ้ายไปจนถึงมุมล่างขวา ทำให้ผู้ใช้มือใหม่สามารถหาตัวอักษรที่ต้องการได้โดยง่าย

ประการที่สาม คำหรือส่วนของคำที่ใช้บ่อยๆ จะอยู่ติด ทำให้สามารถมองหาตัวอักษรตัวต่อไปได้ง่าย รวมทั้ง ขยับมือน้อยลงในการพิมพ์คำเหล่านั้น เช่น ‘the’ หรือ ‘ing’ ซึ่งจะเห็นว่าแป้นตัวอักษรเหล่านั้นอยู่ติดกัน



ภาพแสดงแป้นพิมพ์ ATOMIK แบบหกเหลี่ยม

7. TSwipe [5]

TSwipe เป็นแป้นพิมพ์แบบลากค่านระบบปฏิบัติการแอนดรอยด์ที่รองรับภาษาไทย ซึ่งได้รับความนิยมอย่างสูงในหมู่คนไทยที่ใช้สมาร์ทโฟนแอนดรอยด์ เนื่องจากความแม่นยำในการทำนายคำ โดยการจัดวางตัวแป้นพิมพ์นั้นยังคงมีความใกล้เคียงการจัดวางแป้นพิมพ์คอมพิวเตอร์ กล่าวคือ สำหรับแป้นพิมพ์ภาษานั้นยังคงยึดตามการจัดวางแบบแป้นพิมพ์เกษมณี (แต่ผู้ใช้เองก็สามารถเลือกเปลี่ยนการจัดวางให้เป็นแบบแป้นพิมพ์ปัตตะโชติได้) ซึ่งในจุดเองที่ทำให้ผู้ใช้งานบางส่วนที่ไม่ชินกับการใช้งานคอมพิวเตอร์ไม่สามารถใช้งานได้ทันที แต่จะต้องใช้เวลาเรียนรู้รูปแบบการจัดวางแป้นพิมพ์แบบเกษมณีเสียก่อน จึงจะสามารถใช้งานแป้นพิมพ์ TSwipe ได้อย่างมีประสิทธิภาพ



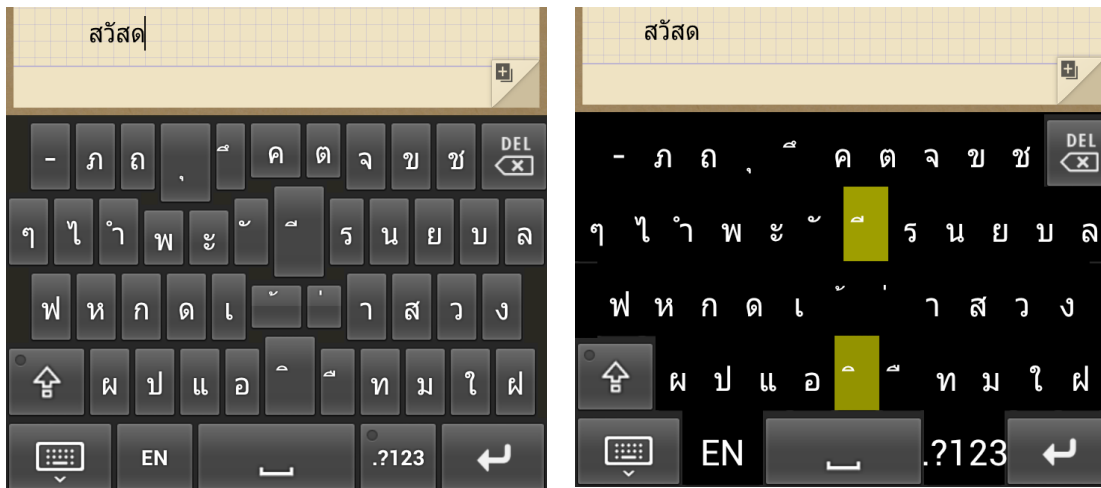
ภาพแสดงแป้นพิมพ์ TSwipe

8. แมนแมน [1]

แมนแมน เป็นอีกทางเลือกหนึ่งของแป้นพิมพ์ไทยบนระบบปฏิบัติการแอนดรอยด์ โดยจุดเด่นของแป้นพิมพ์แมนแมนคือการที่แป้นอักขระแต่ละตัวนั้นจะขยายขนาดขึ้น ตามความความน่าจะเป็นที่แป้นนั้นๆ จะถูกกดเป็นตัวต่อไป

สำหรับผู้ที่ไม่ชอบฟีเจอร์การขยายขนาดแป้นที่ทำให้ตัวแป้นพิมพ์ขาดความสวยงามไป แมนแมนเองก็ยังมีโหมดการเน้นสีแป้นที่มีโอกาสสูงที่จะถูกกดเป็นลำดับต่อไปแทน

อย่างไรก็ตาม ด้วยการที่เน้นหนักไปทางด้านการพิมพ์ ทำให้แมนแมนนั้นเอื้อต่อการใช้งานด้วยสองมือเสียมากกว่า ซึ่งทำให้ขาดความคล่องตัวและความสะดวกสบายในการใช้อันเป็นจุดขายของอุปกรณ์พกพาอย่างสมาร์ทโฟนไป



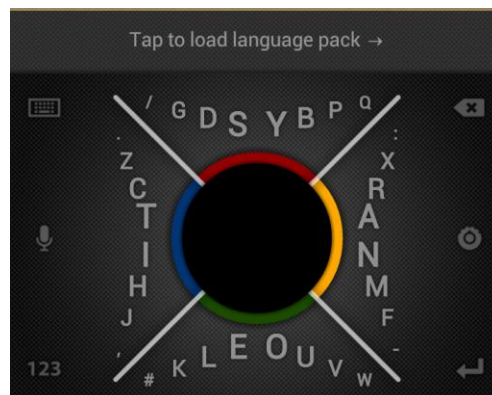
ภาพแสดงแป้นพิมพ์แม่นยำแม่นยำในโหมดขยายแป้น (ซ้าย) และโหมดเน้นสีแป้น (ขวา)

9. 8pen [5]

8pen เป็นแป้นพิมพ์บนระบบปฏิบัติการแอนดรอยด์ที่ไม่ยึดตามรูปแบบแป้นพิมพ์คอมพิวเตอร์ โดย 8pen นั้นถูกพัฒนาขึ้นมาจากแนวคิดที่ว่า แป้นพิมพ์ที่มีลักษณะใกล้เคียงกับแป้นพิมพ์คอมพิวเตอร์นั้น ไม่เหมาะสำหรับการใช้งานบนหน้าจออุปกรณ์พกพาขนาดเล็ก เช่น สมาร์ทโฟน ดังนั้น 8pen จึงได้ออกแบบวิธีการป้อนข้อมูลเข้าเสียใหม่ โดยทำการแบ่งหน้าจอออกเป็น 4 ส่วน และใช้การลากนิ้วเป็นวงกลมล้อมรอบตัวอักษรแต่ละตัวแทนการพิมพ์ ซึ่งการจัดวางตัวอักษรนั้นจะจัดโดยพิจารณาจากความบ่อยในการใช้งานของตัวอักษรนั้นๆ กล่าวคือ ตัวอักษรที่ถูกใช้งานบ่อยในภาษาอังกฤษ เช่น A E I N O S T Y นั้นจะถูกจัดให้อยู่ในตำแหน่งที่สามารถลากนิ้วเลือกได้ง่าย ลากเป็นเส้นสั้นๆ ส่วนตัวอักษรอื่นที่มีการใช้งานค่อนข้างต่ำ เช่น Q X Z ก็จะถูกจัดไว้ในตำแหน่งที่ต้องอาศัยการลากนิ้วที่ยาวกว่าในการพิมพ์ตัวอักษรดังกล่าว

แต่ด้วยการออกแบบวิธีการป้อนข้อมูลแบบใหม่ของ 8pen นี้ ทำให้การจะใช้งาน 8pen ได้อย่างมีประสิทธิภาพนั้นต้องใช้เวลาในการจดจำตำแหน่งของตัวอักษรแต่ละตัวให้ได้เสียก่อน โดยทางผู้พัฒนาก็ได้มีการพัฒนาเกมสำหรับฝึกพิมพ์คำด้วย 8pen ขึ้นเพื่อให้การเรียนรู้การใช้งานเจ้า 8pen นั้นมีความสนุกสนานมากยิ่งขึ้น

แต่อย่างไรก็ตาม ในปัจจุบันนี้ 8pen ยังคงไม่รองรับภาษาไทย



ภาพแสดง 8pen

10. MessageEase [5]

MessageEase เป็นแป้นพิมพ์ออกแบบมาเพื่อลดจำนวนแป้นตัวอักษรบนหน้าจอ ซึ่งทำให้ขนาดของแป้นแต่ละแป้นนั้นมีขนาดใหญ่ขึ้น โดย MessageEase ทำการนำตัวอักษรที่ใช้บ่อย 9 ตัวมาเป็นตัวอักษรประจำแป้นตัวอักษรแต่ละแป้น อันได้แก่ a n i h o r t e s ซึ่งสามารถกดที่แป้นเพื่อพิมพ์ตัวอักษรนั้นๆ ได้เลย ในขณะที่ตัวอักษรอื่นๆ จะถูกใส่ไว้รอบๆ ตัวอักษรหลักของแต่ละแป้นแทน และใช้การลากนิ้วไปยังมุมต่างๆ ของแป้นเพื่อเลือกพิมพ์ตัวอักษรนั้นๆ

โดยแนวคิดในการลดจำนวนแป้นลงนั้นเป็นแนวคิดที่น่าสนใจมาก อย่างไรก็ตาม ในปัจจุบันนี้ แป้นพิมพ์ MessageEase ยังไม่รองรับภาษาไทย

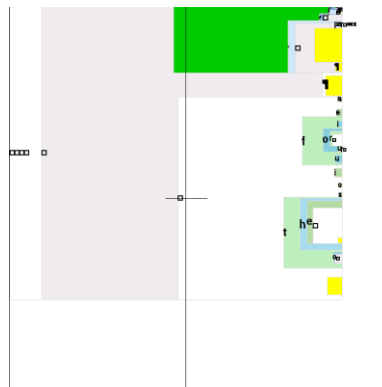


ภาพแสดง MessageEase

11. Dasher [5]

Dasher เป็นอีกหนึ่งวิธีการป้อนข้อมูลเข้าระบบปฏิบัติการแอนดรอยด์ที่ได้ถูกออกแบบใหม่ โดยไม่ยึดติดอยู่กับรูปแบบแป้นพิมพ์คอมพิวเตอร์ โดยการป้อนข้อมูลนั้นจะเป็นในลักษณะของการลากจุดกลางของ Dasher ให้เข้าไปยังบริเวณของตัวอักษรแต่ละตัว โดยจะตัวอักษรที่ถูกใช้มาก หรือมีโอกาสที่จะถูกพิมพ์ต่อไปสูง ก็จะมีพื้นที่ใหญ่ขึ้นตามไปด้วย

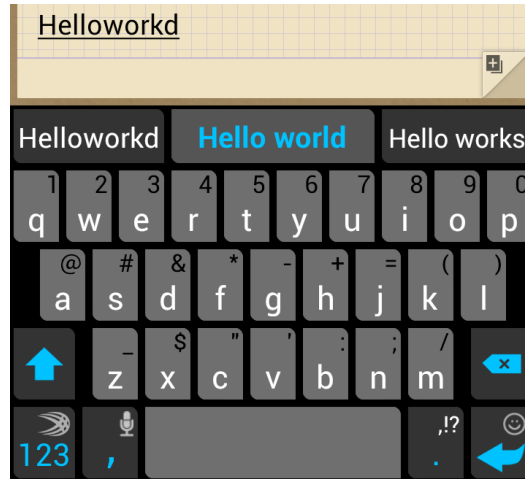
อย่างไรก็ตาม จากการทดลองใช้ พบว่าการที่จะใช้ Dasher ให้ได้อย่างมีประสิทธิภาพนั้น ต้องอาศัยความสามารถในการมอง และสมาธิสูงระดับหนึ่ง เนื่องจากในการเลื่อนจุดกลางเข้าไปยังพื้นที่ตัวอักษรนั้นเป็นไปด้วยความเร็วที่ค่อนข้างสูง (แม้จะตั้งค่าให้เลื่อนด้วยความเร็วต่ำสุดแล้วก็ตาม)



ภาพแสดง Dasher

12. SwiftKey [5]

SwiftKey นั้นเป็นแป้นพิมพ์ภาษาอังกฤษบนระบบปฏิบัติการแอนดรอยด์ ซึ่งมีการจัดวางแป้นพิมพ์แบบ QWERTY คือ มีการจัดวางแป้นพิมพ์เหมือนกับแป้นพิมพ์คอมพิวเตอร์ แต่สิ่งที่ทำให้ SwiftKey โดดเด่นกว่าแป้นพิมพ์อื่นๆ นั้น คือ SwiftKey มีการเดาและตัดแบ่งคำศัพท์ที่ดี

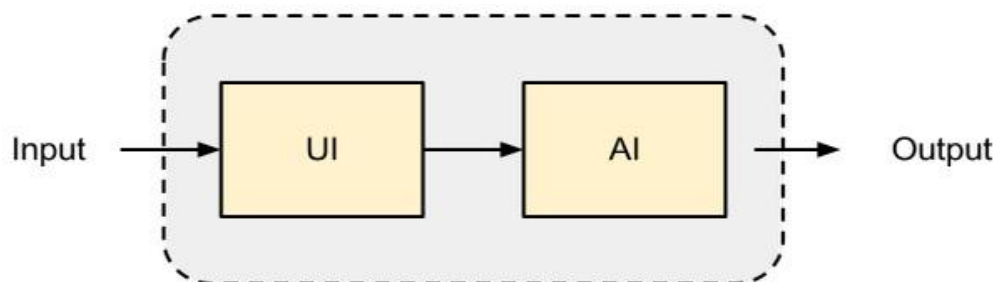


ภาพแสดง SwiftKey

จะเห็นว่า SwiftKey มีการแบ่งสิ่งที่พิมพ์ออกเป็น 2 คำย่อย คือ 'Hello' และ 'world' นั่นคือการเดาและตัดแบ่งคำของ SwiftKey นั้นสามารถทำได้กับสายตัวอักษรที่ประกอบขึ้นจากคำหลายคำได้ ทำให้ผู้ใช้สามารถพิมพ์ได้อย่างรวดเร็วโดยไม่ต้องคอยพะวงกับการเว้นวรรคคำหรือการกดให้ถูกแป้น

เช่นเดียวกับแป้นพิมพ์แม่นยำ เนื่องจาก SwiftKey นั้นถูกจัดวางในลักษณะเดียวกับแป้นพิมพ์คอมพิวเตอร์ ซึ่งการใช้ SwiftKey ให้มีประสิทธิภาพสูงสุดนั้น ผู้ใช้ต้องหยิบจับอุปกรณ์ของตนด้วยมือทั้งสองมือ ทำให้การใช้สมาร์ตโฟนนั้นไม่คล่องตัวอย่างที่ควรจะเป็น

แนวคิด



ระบบจะแบ่งออกเป็นสองส่วนย่อย ดังนี้

ส่วนที่หนึ่ง UI ในที่นี้คือ Layout ที่ใช้รับข้อมูลที่ผู้ใช้งานเข้ามา โดย Layout ที่ออกแบบจะวัดประสิทธิภาพโดยดูจากเวลาที่ใช้ในการเรียนรู้แรกเริ่ม (Learning Curve) และ ระยะในการลากเพื่อพิมพ์คำต่างๆ โดยตัววัดทั้งสองตัวยังมีค่าน้อยยิ่งดี

ส่วนที่สอง AI ในที่นี้ประกอบไปด้วยระบบการแก้ไขคำผิด(Word Correction System) และ ระบบทำนายคำ (Word Prediction System) วัดประสิทธิภาพโดยนับจำนวนของคำที่ระบบทำนายแล้วถูกใช้โดยผู้ใช้ (ยิ่งมากยิ่งมีประสิทธิภาพ)

เครื่องมือที่ใช้ในการพัฒนา

ภาษาโปรแกรม: ภาษา Java

เครื่องมือ: Eclipse IDE, Android SDK

รายละเอียดของระบบที่จะพัฒนา (System Specification)

Input/Output Specification

Input – ตำแหน่ง ลำดับ และทิศทางในการลากนิ้วบนแป้นพิมพ์ของผู้ใช้

Output – คำที่มีความน่าจะเป็นสูงที่สุดที่จะเป็นคำที่ผู้ใช้ต้องการ และรายการของคำใกล้เคียง

Functional Specification

- ซอฟต์แวร์สามารถทำงานบนระบบปฏิบัติการแอนดรอยด์ได้
- ซอฟต์แวร์สามารถรับข้อมูลจากการลากนิ้วบนบริเวณแป้นพิมพ์ของผู้ใช้ได้
- ซอฟต์แวร์สามารถให้คำที่แม่นยำ ตรงกับความต้องการของผู้ใช้ได้
- ซอฟต์แวร์สามารถเก็บบันทึกความถี่ในการใช้งานคำศัพท์ต่างๆ เพื่อนำมาประกอบการตัดสินใจในการแนะนำ

หรือทำนายคำที่ผู้ใช้กำลังจะพิมพ์ได้

งานวิจัยเบื้องต้น

1. ศึกษารูปแบบของการใช้ภาษาไทยในบริการต่างๆ บนสมาร์ตโฟน เช่น อีเมล, instant messaging เป็นต้น ไม่ว่าจะเป็นตัวอักษร สระ และ วรรณยุกต์ที่ใช้บ่อย คำที่ใช้บ่อย คู่คำที่ใช้บ่อย
2. ศึกษาหลักการจัดวางแป้นพิมพ์แบบต่างๆ โดยเปรียบเทียบกับของที่มีอยู่แล้วใน Google Play Store
3. ศึกษาหลัก Ergonomics ของคนในขณะที่ใช้สมาร์ตโฟนด้วยมือเดียว
4. ศึกษาการพัฒนาซอฟต์แวร์บนแอนดรอยด์

วัตถุประสงค์

1. เพื่อพัฒนารูปแบบวิธีการป้อนข้อมูลเข้าภาษาไทยสำหรับแอนดรอยด์
2. เพื่อพัฒนาให้คนไทยสามารถใช้สมาร์ตโฟนแอนดรอยด์ได้สะดวกมากขึ้น
3. เพื่อพัฒนาทักษะการพัฒนาแอปพลิเคชันบนแอนดรอยด์
4. เพื่อพัฒนาทักษะการออกแบบแอปพลิเคชันโดยมุ่งเน้นที่การมอบประสบการณ์การใช้งานที่ดีที่สุดให้กับผู้ใช้

ขอบเขตของโครงการ

โครงการ “ทัมบัสไวป์: คีย์บอร์ดภาษาไทยอัจฉริยะใช้งานง่ายสำหรับแอนดรอยด์” มีเป้าหมายหลักคือ พัฒนารูปแบบการป้อนข้อมูลเข้าภาษาไทยบนแอนดรอยด์ เพื่ออำนวยความสะดวกกับผู้ใช้งานบางกลุ่มที่ไม่คุ้นเคยกับแป้นพิมพ์เกษมณี

รูปแบบการใช้งานหลักนั้นจะใช้การลาก (swipe) โดยเมื่อผู้ทำการลากนิ้วไปตามลำดับตัวอักษรตามคำที่ผู้ใช้ต้องการ

จะป้อน ตัวแอปพลิเคชันจะทำการวิเคราะห์ลำดับของตัวอักษรตามบริเวณของปุ่มที่นิ้วของผู้ใช้ลากผ่าน แล้วคืนคำที่มีความน่าจะเป็นที่ผู้ใช้จะหมายถึงคำดังกล่าวสูงที่สุด พร้อมทั้งให้รายการของคำที่มีความน่าจะเป็นที่จะเป็นคำที่ผู้ใช้ต้องการค่อนข้างสูง เพื่อรองรับคำอื่นๆ ที่มีความใกล้เคียงกันในกรณีที่ผู้ใช้งานลากนิ้วผิดตำแหน่ง

ขั้นตอนการดำเนินงาน

1. กำหนดปัญหา ขอบเขต วัตถุประสงค์ และประโยชน์ในการทำโครงการ
 2. ศึกษาข้อมูลเกี่ยวกับทฤษฎี และงานวิจัยที่เกี่ยวข้องในการทำโครงการ ได้แก่
 - การแบ่งคำไทย
 - การออกแบบส่วนต่อประสานกับผู้ใช้ และประสบการณ์การใช้งาน
 - ปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง
 - การยศาสตร์
 3. เสนอโครงร่างให้อาจารย์ที่ปรึกษาพิจารณา
 4. ศึกษาข้อมูลรูปแบบการใช้คำภาษาไทยจาก e-mail, social network และ instant messaging ซึ่งเป็นสิ่งที่ผู้ใช้งานสมาร์ทโฟนใช้กันมาก
 5. ออกแบบการจัดวางแป้นพิมพ์ และทดสอบการใช้งานเบื้องต้นเพื่อหารูปแบบการจัดวางที่ใช้งานง่าย ใช้เวลาเรียนรู้ต่ำ
 6. ออกแบบการทำนายคำ และการวางตำแหน่งตัวอักษร
 7. พัฒนาแป้นพิมพ์ภาษาไทยบนแอนดรอยด์ตามการจัดวางแป้นพิมพ์ที่ได้ออกแบบไว้
 8. ทดสอบประสิทธิภาพการทำงานของแป้นพิมพ์โดยการให้ผู้ใช้งานสมาร์ทโฟนแอนดรอยด์จำนวนหนึ่งทดลองใช้งาน
- ก่อน
9. นำเสนอผลการปฏิบัติงานต่ออาจารย์ที่ปรึกษา

ประโยชน์ที่คาดว่าจะได้รับในเชิงความรู้คอมพิวเตอร์และการใช้ประโยชน์ด้านต่างๆ

ผู้ใช้สมาร์ทโฟนแอนดรอยด์สามารถป้อนข้อมูลเข้าภาษาไทยได้อย่างมีประสิทธิภาพ โดยใช้เวลาการเรียนรู้ต่ำ เทียบกับแป้นพิมพ์ที่มีอยู่ในตลาดทั่วไปโดยวัดจากเวลาที่ใช้ในการเรียนรู้เริ่มต้นและ ความเร็วในการพิมพ์ในหน่วยของจำนวนคำต่อนาที (wpm)

รายการอ้างอิง

ภาษาไทย

- [1]ณัฐ ปิยะปราโมทย์. "Keyboard ManMan." [ออนไลน์].
เข้าถึงได้จาก: <http://nattster.siamdev.net/2011/01/keyboard-manman/> 2554. สืบค้น 7 กันยายน 2555.
- [2]ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. หน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา. "การวัดเปรียบเทียบสมรรถนะเพื่อพัฒนามาตรฐานการประมวลผลภาษาไทย." [ออนไลน์].
เข้าถึงได้จาก: <http://thailang.nectec.or.th/best/?q=node/4> [ม.ป.ป.]. สืบค้น 10 สิงหาคม 2555.

ภาษาอังกฤษ

- [3]Anirudha Joshi and others. Design and evaluation of Devanagari virtual keyboards for touch screen mobile phones. 2011
- [4]Associated Press. "Worldwide market share for smartphones, a market dominated by Apple and Android." [Online]. Available: http://www.washingtonpost.com/business/technology/worldwide-market-share-for-smartphones-a-market-dominated-by-apple-and-android/2012/09/06/93154dfe-f834-11e1-a93b-7185e3f88849_story.html 2012. Retrieved September 8, 2012
- [5]Darren Meehan. "15 Awesome Keyboards for Android." [Online]. Available: <http://android.appstorm.net/roundups/productivity-roundups/15-awesome-keyboards-for-android/> 2011. Retrieved September 7, 2012
- [6]I. Scott MacKenzie. Using paper mockups for evaluating soft keyboard layouts. 2007
- [7]Jean D. Ichbiah. Method for designing an ergonomic one-finger keyboard and apparatus therefor. 1996
- [8]Jochen Rick. Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. 2010
- [9]Shumin Zhai. "ATOMIK and other stylus keyboards." [Online]. Available: <http://www.almaden.ibm.com/u/zhai/ATOMIK.htm> [n.d.]. Retrieved September 7, 2012