

ความรู้เบื้องต้นภาษา Lua และ Corona SDK  
Introduction to Lua and Corona SDK

นาย กนก เบญจบุญยฤทธิ์ 5410513002

นาย ไตรรงค์ แซ่จิว 5410513025

นาย ศิรสิทธิ์ ฟุ้งชูเกียรติ 5410513064

อาจารย์ที่ปรึกษา

ดร. จันทวรรณ ปิยะวัฒน์

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา 460-491 สัมมานา  
 ภาคเรียนที่ 1 ปีการศึกษา 2557  
คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

คำนำ

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชาสัมมานาระดับชั้นปีการศึกษาที่ 4 เพื่อให้ได้ศึกษาหาความรู้ในเรื่องของภาษา Lua และ Corona SDK เบื้องต้น เช่น คำสั่งพื้นฐาน ของภาษา Lua , การใช้ Lua กับ Corona SDK เบื้องต้น ,ตัวอย่างแอพพลิเคชั่นบนมือถือ ที่ใช้ Lua และ Corona และได้ศึกษาอย่างเข้าใจเพื่อเป็นประโยชน์กับการศึกษา

ผู้จัดทำหวังว่า  รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา ที่กำลัง หาข้อมูลเรื่องนี้อยู่ หากมีข้อแนะนำหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

 วันที่ 30 ตุลาคม 2557

สารบัญ

เรื่อง                                                                       หน้า

คำนำ                                                                                ก

สารบัญ                                                                                       ข

สารบัญภาพ ค

บทนำ                      1

สารบัญรูป

เรื่อง                                                                       หน้า

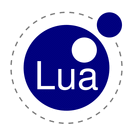
คำนำ                                                                                 ก

สารบัญ                                                                                      ข

บทที่ 1

บทนำ

1.ภาษา Lua



รูปที่ 1.1 โลโก้ Lua

* 1. ที่มาและความหมาย

Lua ถูกออกแบบมาการใช้งานและการพัฒนาโดยทีมงาน PUC-Rio, the Pontifical Catholic University of Rio de Janeiro ในประเทศบราซิล Lua เกิดและการพัฒนาใน Tecgraf เดิมกลุ่มเทคโนโลยีคอมพิวเตอร์กราฟิกของ PUC-Rio แต่ในปัจจุบันเป็นที่ตั้งของ LabLua, ห้องปฏิบัติการของภาควิชาวิทยาการคอมพิวเตอร์ของ PUC-Rio  
 "Lua" ออกเสียง LOO-Ah หมายถึง "ดวงจันทร์" ในภาษาโปรตุเกส ไม่ได้เป็นตัวย่อหรือย่อแต่คำนาม โดยเฉพาะอย่างยิ่ง "Lua" เป็นชื่อชื่อของดวงจันทร์ของโลกและชื่อของภาษา   
ชื่อส่วนใหญ่ผู้คนจะเขียนกันในภาษาที่ไม่สุภาพคือ "LUA" กรุณาอย่าเขียนว่า "LUA" ซึ่งเป็นทั้งน่าเกลียดและทำให้เกิดความสับสนแล้วเพราะมันจะกลายเป็นตัวย่อที่มีความหมายที่แตกต่างกันสำหรับคนที่แตกต่างกัน ดังนั้นควรเขียน "Lua" ที่ถูกต้อง

1.2. ทำไมต้องเลือก Lua

1.2.1 Lua มีพื้นที่ขนาดเล็ก

เพิ่ม Lua เพื่อการประยุกต์ใช้ไม่ได้ขยายตัวมัน Tarball สำหรับ Lua 5.2.3 ซึ่งมีรหัสแหล่งที่มาและเอกสารจะถูกบีบอัดและ 246K 960K ไม่มีการบีบอัด แหล่งที่มามีประมาณ 20,000 สายของซีภายใต้ Linux, ล่าม Luaสร้างขึ้นด้วยทุกห้องสมุด Lua มาตรฐานใช้ 182K และห้องสมุด Lua จะ 244K  
 1.2.2 Lua เป็นภาษาที่ประสิทธิภาพ และ เป็นภาษาที่ผ่านการพิสูจน์แล้ว

Lua ได้ถูกนำมาใช้ส่วนของโปรแกรมประยุกต์หลายโปรแกรม เช่น ส่วนของ Adobe Photoshop Lightroom โดยเน้นระบบประมวลผล ที่ใช้ชิปหรือไมโครโพรเซสเซอร์ที่ออกแบบมาโดยเฉพาะ เป็นระบบคอมพิวเตอร์ขนาดจิ๋วที่ฝังไว้ในอุปกรณ์ ระบบ เช่นตัวกลาง Ginga สำหรับโทรทัศน์ดิจิตอลในบราซิล และเกมชื่อดัง เช่น World of Warcraft และเกมส์ Angry Birds

Lua ในปัจจุบันคือภาษาสคริปต์ชั้นนำในเกม Lua มีคู่มืออ้างอิงและมีหนังสือหลายเล่มเกี่ยวกับเรื่องนี้ หลายรุ่น Lua ได้รับการเผยแพร่และนำมาใช้ในการใช้งานจริงนับตั้งแต่การสร้างในปี 1993 Lua ที่ใช้โดดเด่นใน HOPL III, The Third ACM SIGPLAN เป็นประวัติการเขียนโปรแกรมภาษาในการประชุม ในเดือน มิถุนายน 2007

1.2.3 Lua ได้อย่างรวดเร็ว

Lua มีชื่อเสียงเป็นสมควรจะสำหรับการทำงาน อ้างจากคำว่า "as fast as Lua" เป็น มันเป็นความต้องการที่ภาษาสคริปต์อื่นๆ อาจจะมีทดสอบจากหลากหลายของมาตรฐานแสดงให้เห็นว่า Lua เป็นภาษาที่เร็วที่สุดในขอบเขตของภาษาสคริปต์ตีความ Luaได้อย่างรวดเร็วไม่เพียง แต่ในโปรแกรมมาตรฐานปรับแต่ง, แต่ในการทำงานในชีวิตจริงด้วย ส่วนที่สำคัญของการใช้งานขนาดใหญ่ได้รับการเขียนใน Lua ถ้าคุณต้องการความเร็วมากขึ้นลอง LuaJIT, การดำเนินการเป็นอิสระจาก Lua สามารถแสดงการทำงานได้ ณ ตอนการทำงานเลย

1.2.4 Lua ใช้งานสะดวก

Lua เป็นการประมวลข้อมูลในแพ็คเกจขนาดเล็กและสร้างออกจากกล่องในแพลตฟอร์มทั้งหมดที่มีมาตรฐานภาษา C Lua สามารถทำงานบน Unix และ Window, บนอุปกรณ์มือถือ  (ที่ใช้ Android, iOS, BREW, Symbian, Windows Phone) สำหรับไมโครโปรเซสเซอร์ฝังตัวบน IBM เมนเฟรมคอมพิวเตอร์ ฯลฯ

1.2.5 Lua สมองกลฝังตัว

Lua เป็นเครื่องมือภาษาได้อย่างรวดเร็วด้วยขนาดเล็กที่คุณสามารถใช้งานได้ง่ายในการใช้งาน Lua มีระบบ API ที่ง่ายและดีที่ช่วยให้บูรณาการที่ดีกับโค้ดที่เขียนในภาษาอื่น ๆ มันเป็นเรื่องง่ายที่จะขยาย Lua บูรณาการใช้กับในภาษาอื่น ๆ นอกจากนี้ยังเป็นเรื่องง่ายที่จะโปรแกรมที่เขียนในภาษาอื่น ๆ ที่มี Lua  Lua ได้รับการใช้ในการขยายโปรแกรมที่เขียนไม่เพียง แต่ใน C และ C ++ แต่ยังอยู่ใน Java, C #, Smalltalk Fortran, Ada, Erlang และแม้กระทั่งในการเขียนสคริปต์ภาษาอื่น ๆ เช่น Perl และRuby

1.2.6 Lua ฟรี

Lua เป็นซอฟต์แวร์โอเพนซอร์สฟรีแจกจ่ายภายใต้ใบอนุญาตเสรีนิยมมาก (ใบอนุญาต MIT ที่รู้จักกันดี) มันอาจจะถูกใช้เพื่อวัตถุประสงค์ใด ๆ รวมทั้งวัตถุประสงค์ในเชิงพาณิชย์ได้อย่างไม่มีค่าใช้จ่าย เพียงดาวน์โหลดและใช้งานได้

2. Corona SDK



รูปที่ 1.2 โลโก้ Corona SDK

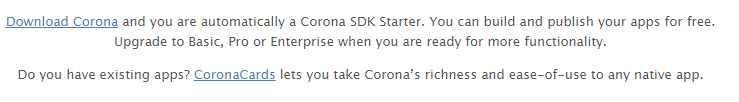
Corona SDK คือ SDK อาจเป็นพวก API (application programming interface) ซึ่งเป็นโค้ดแบบสำเร็จรูป ให้เรานำมาใช้ได้ต่อยอดได้ ที่ใช้สำหรับการพัฒนาแอพพลิเคชั่นบนมือถือในระบบ iOS และ android Corona SDK จะใช้ Lua ในการเขียนโปรแกรม ซึ่งง่าย และรวดเร็วในการสร้างแอพพลิเคชั่น โดยใช้ OpenGL ในการสร้างกราฟฟิคต่างๆ Corona SDK มี API ที่มอบความสะดวกรวดเร็วให้แก่ผู้ใช้งาน งานอย่าง animating objects สามารถข้ามแพลตฟอร์มได้  
การเขียนโปรแกรมเพียงครั้งเดียว สามารถทำการใช้ได้ทั้งในระบบ android, NOOK, Kindle Fire และ iOS

บทที่ 2

ขั้นตอนการติดตั้ง Corona SDK และ Lua

2.1 ขั้นตอนการติดตั้ง Corona SDK มีดังนี้

ดาวโหลด เวอร์ชั่นฟรีได้ ที http://coronalabs.com/products/corona-sdk/



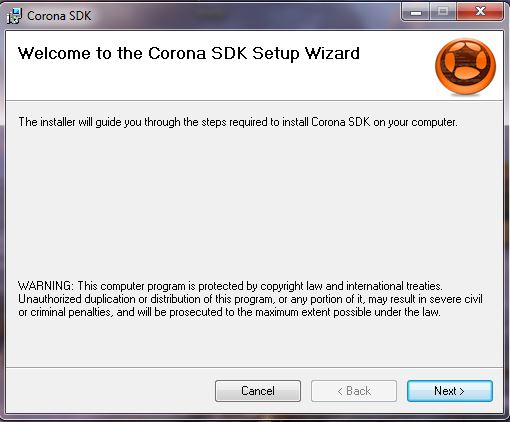
รูปที่ 2.1 แสดงตำแหน่งการโหลด Corona SDK ฟรี

1.เปิดโปรแกรมตัวติดตั้ง Corona SDK เลือก I Agree จากนั้น กด Nextไปขึ้นตอนถัดไป



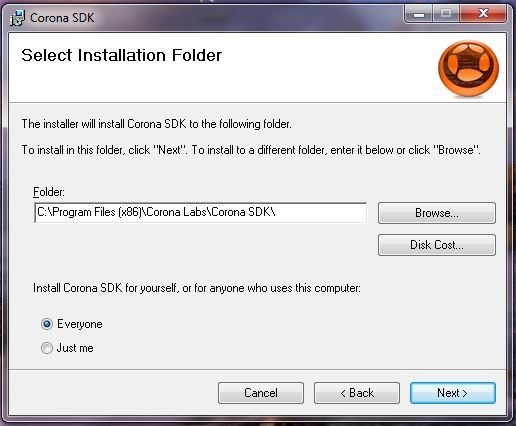
รูปที่ 2.2 แสดงการขั้นตอนการยอมรับเงื่อไข

2.จากนั้นทำการกด Next เพื่อไปขั้นตอนถัดไป



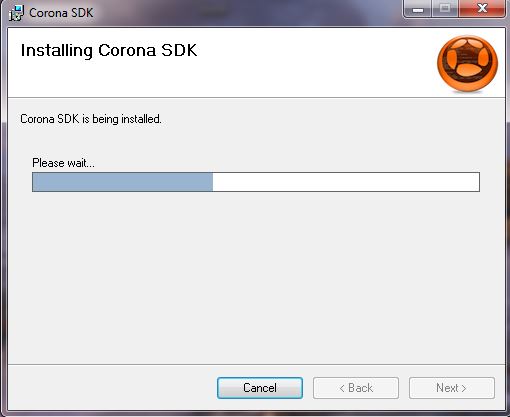
รูปที่ 2.3 แสดงการขั้นตอนการติดตั้ง

3.ทำการเลือกโฟลเดอร์ที่จัดเก็บข้อมูล และเลือกการเข้าใช้เป็น Everyoneใช้ได้ทุกหรือ การตั้งใช้ส่วน บุคคล



รูปที่ 2.4 แสดงการขั้นตอนการเลือกตำแหน่งที่เก็บไฟล์โปรแกรม

5.รอการติดตั้งโปรแกรม



รูปที่ 2.5 แสดงการรอติดตั้งโปรแกรมลงเครื่อง

8.เมื่อติดตั้งโปรแกรมเสร็จหน้าตาของโปรแกรมที่ทำการติดตั้ง

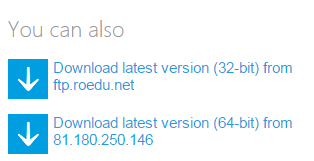


รูปที่ 2.6 แสดงไอคอนของโปรแกรม

2.2 ขั้นตอนการติดตั้ง Lua

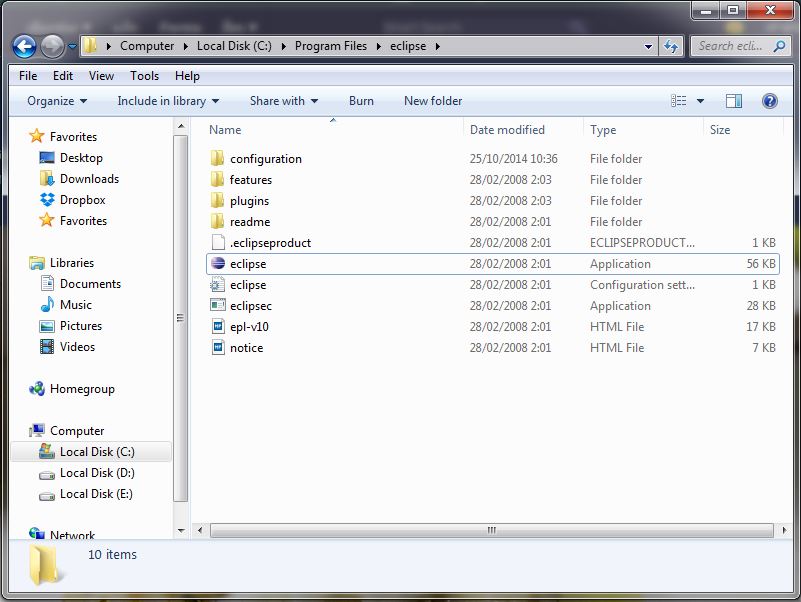
ทำการดาวน์โหลด eclipse สำหรับ version นักพัฒนาโปรแกรม เลือกโหลดให้ตรงกับเวอร์ชั่น ของ ประปฏิบัติการของตัวเอง

http://eclipse-ide-for-java-ee-developers.software.informer.com/3.3/



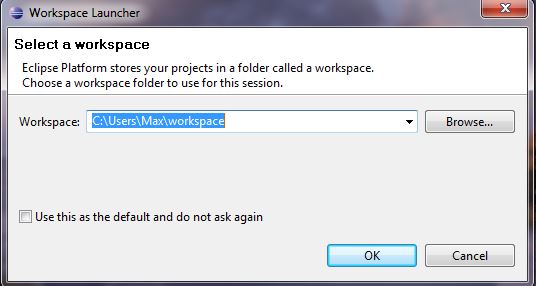
รูปที่ 2.7 แสดงเลือกเวอร์ชั่นระบบปฏิบัติการ

1.ทำการแตกไฟล์และเปิดโฟลเดอร์ Eclipse ให้คลิกเปิดโปรแกรม Eclipse



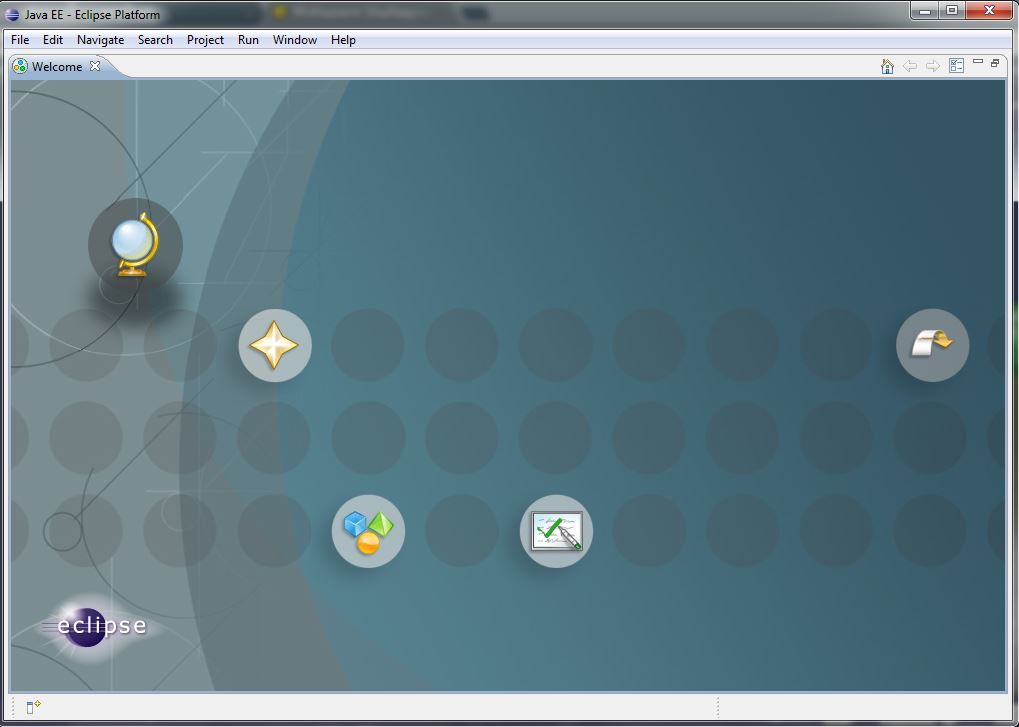
รูปที่ 2.8 แสดงหน้าโฟลเดอร์โปรแกรม eclipse

2.ทำการเลือกโฟลเดอร์ที่จะเก็บไฟล์



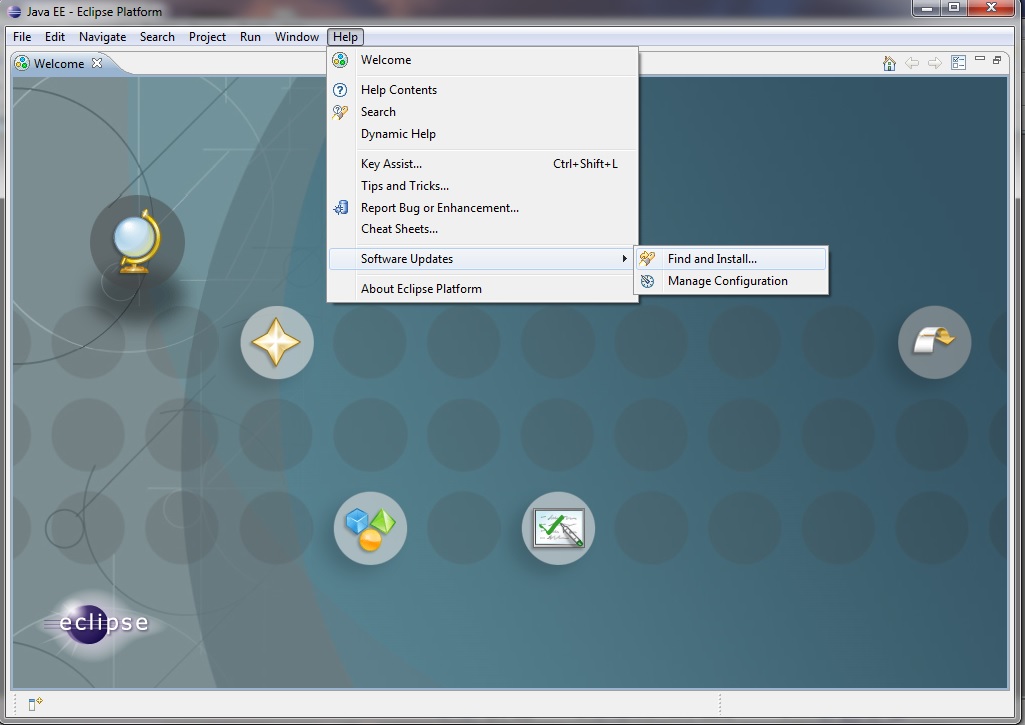
รูปที่ 2.9 แสดงการขั้นตอนการเลือกตำแหน่งที่เก็บไฟล์โปรแกรม

3.หน้าโปรแกรม Eclipase Plantform



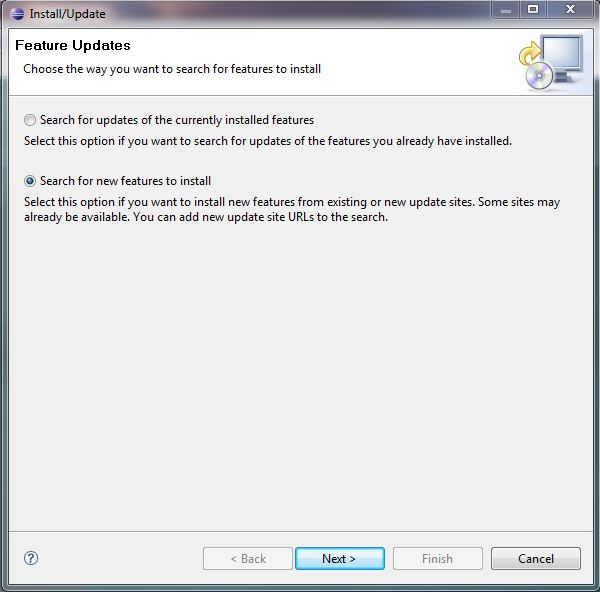
รูปที่ 2.10 แสดงหน้าต่างแรกของโปรแกรม Eclipase Plantform

4. ทำการกด Help เลือก Software Updates > Find and Install



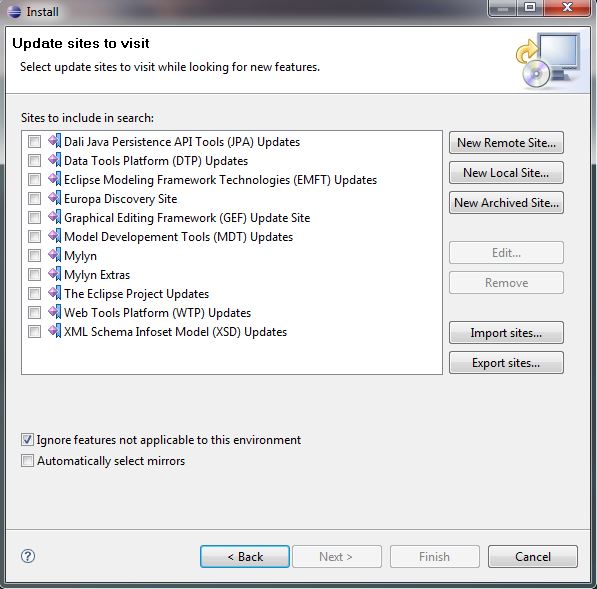
รูปที่ 2.11 แสดงการค้นหาติดตั้งภาษาเสริม

5.ทำการเลือก Search for new features to install แล้วกด Next



รูปที่ 2.12 แสดงขั้นตอนการค้นหาฟีเจอร์

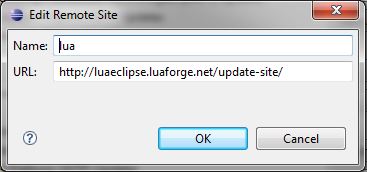
6. ทำการเลือก New Local site



รูปที่ 2.13 แสดงขั้นตอนการเพิ่มภาษา Lua

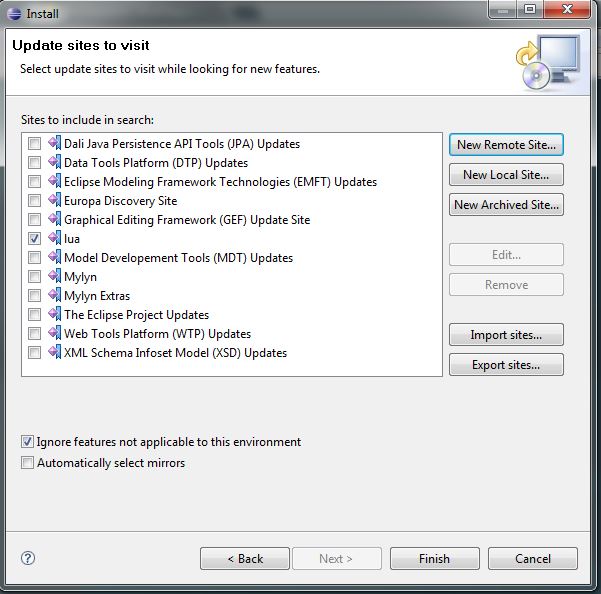
7.ให้ใส่ Name : lua

URL : [http://luaeclipse.luaforge.net/update-site/ แล้ว](http://luaeclipse.luaforge.net/update-site/%20แล้ว) กด OK



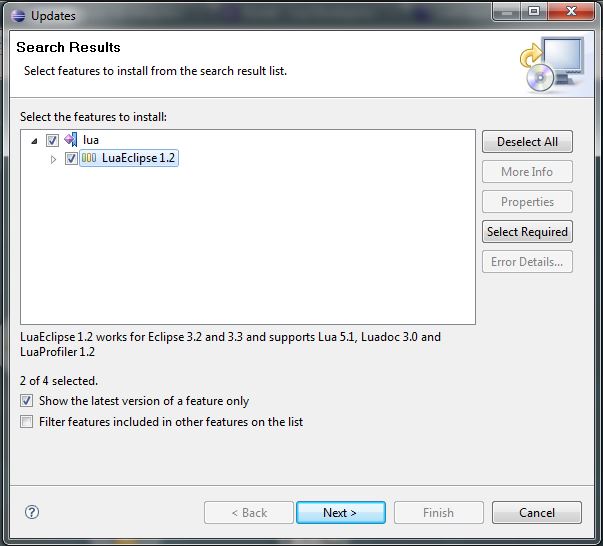
รูปที่ 2.14 แสดงขั้นตอนใส่ URL ภาษา Lua

8.ให้กดเครื่องหมายถูก หน้า lua แล้วกด Finish



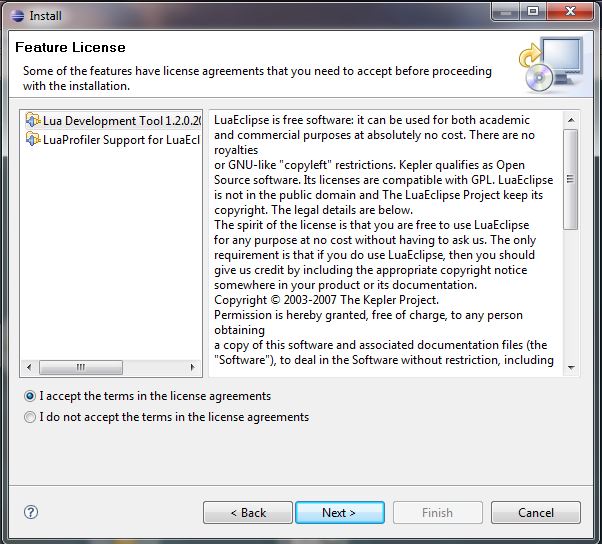
รูปที่ 2.15 แสดงขั้นตอนการเลือก Lua

9.ให้ทำการกดเครื่องหมายถูกที่ Lua แล้วจากนั้นเลือก Show the latest version of a feature only แล้วกด Next



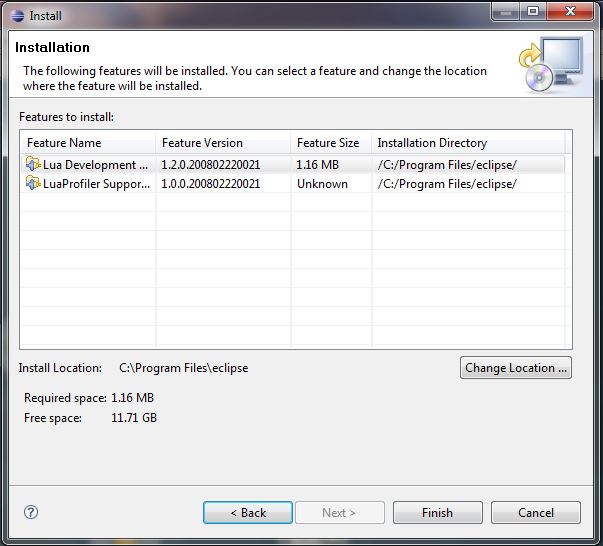
รูปที่ 2.16 แสดงขั้นตอนการเลือกเวอร์ชั่น Lua

10. ให้กดเลือก I accept the terms in the license agreement ตกลงยอมรับ เงื่อนไข แล้วกด Next



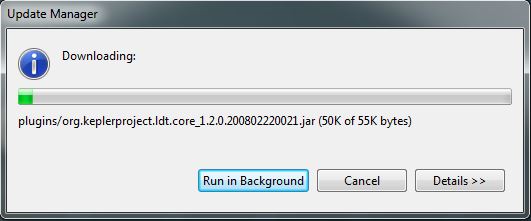
รูปที่ 2.17 แสดงขั้นตอนการตกลงยอมรับเงื่อไข Lua

11.ตรวจสอบ Lua ที่จะลง เวอร์ชั่น และ พื้นที่จัดเก็บ เมื่อถูกต้องแล้วให้ทำการกด Finish เพื่อเป็นการยืนยัน



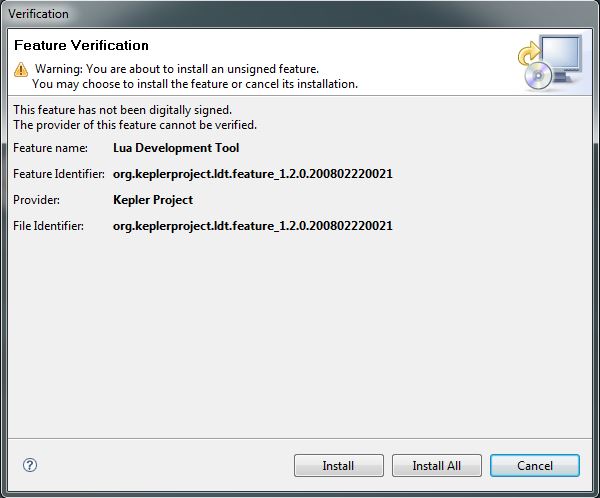
รูปที่ 2.18 แสดงขั้นตอนยืนยันลง Lua

12.รอการอัพโหลดตัวโปรแกรม



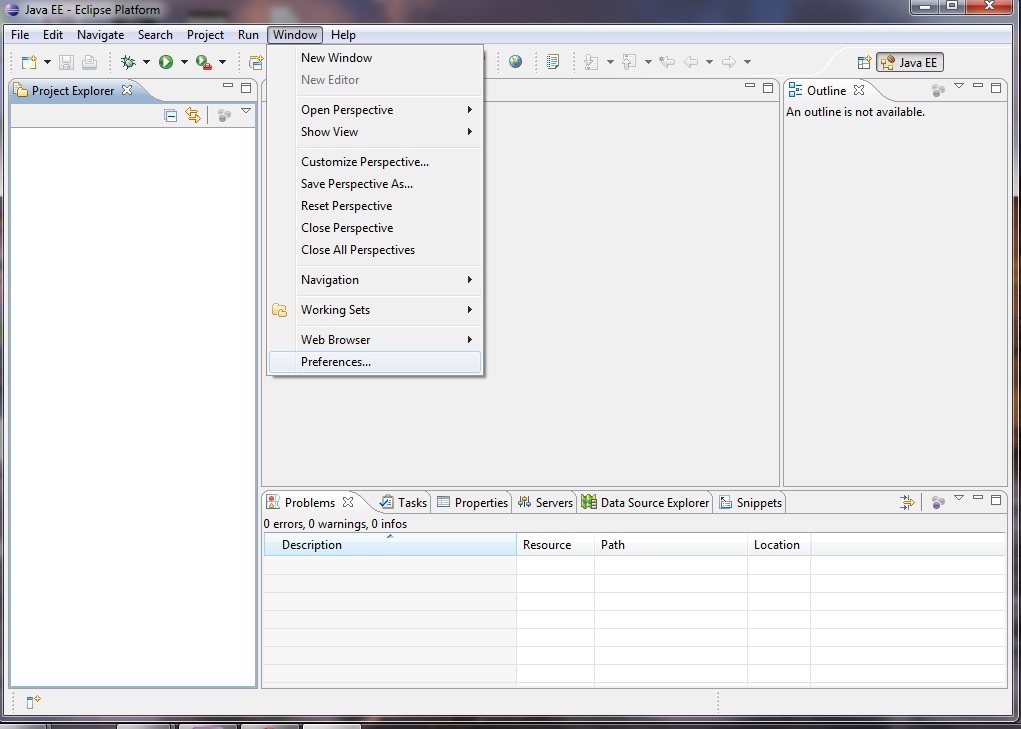
รูปที่ 2.19 แสดงขั้นตอนข้อมูลการลง

13.ให้กดคลิก Install All ตกลงที่จะลงโปรแกรมทั้งหมด



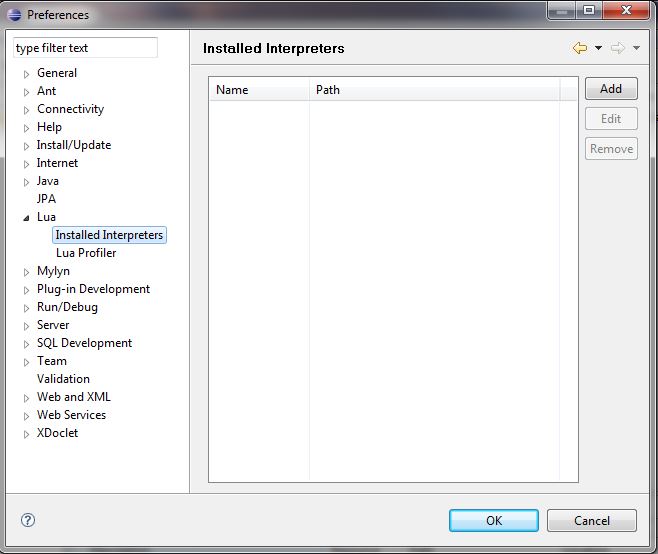
รูปที่ 2.20 แสดงขั้นตอนการเลือกลงโปรแกรม

15.จะขึ้นหน้าโปรแกรม Eclipse Plantform ให้เลือก Windows > Preferences



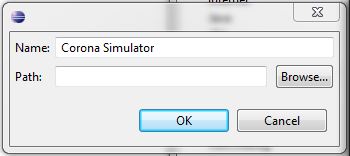
รูปที่ 2.21 แสดงขั้นตอนการเพิ่ม

16.ให้เลือกที่ Lua > installed Interpreters > Add



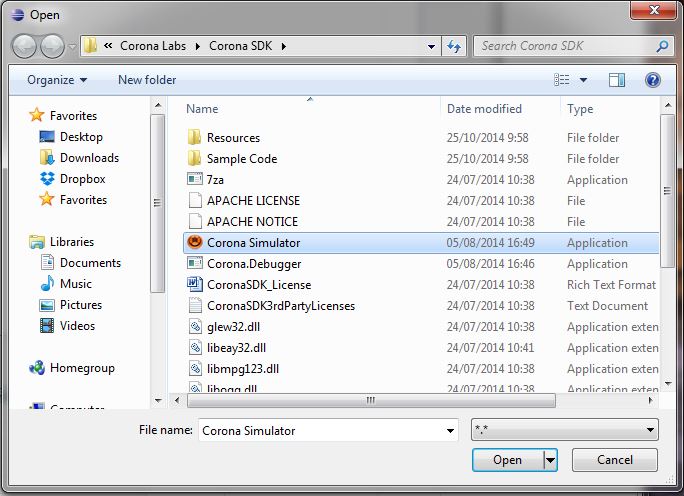
รูปที่ 2.22 แสดงขั้นตอนการเลือกลงโปรแกรม

17.ให้พิมพ์ Name : Corona Simulater > กดคลิก Browse



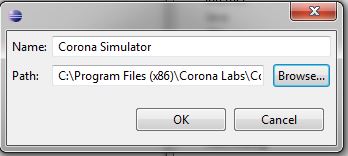
รูปที่ 2.23 แสดงการค้นหา Corona Simulater

18. ให้เลือกโปรแกรม Corona Simulator ดังรูป



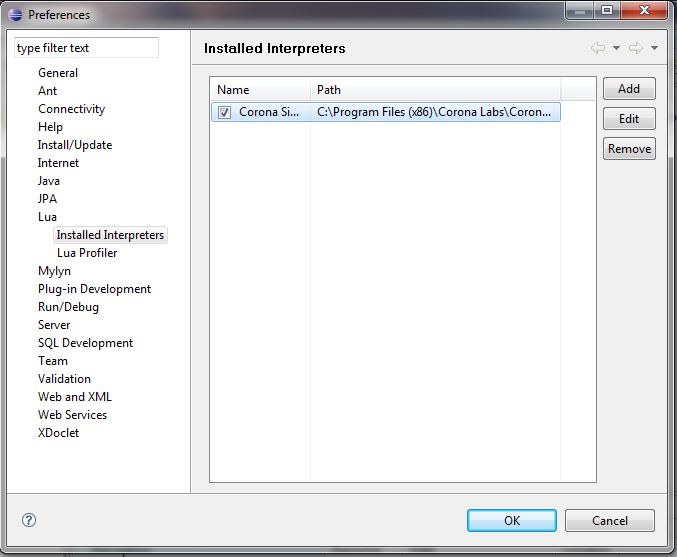
รูปที่ 2.24 แสดงการเลือก Corona Simulater

19. ทำการกด OK



รูปที่ 2.25 แสดงการตกลงเลือก Corona Simulater

20.ให้กดเครื่องหมายถูก และ กด OK ก็จะทำการติดตั้งภาษา Lua เสร็จ



รูปที่ 2.26 แสดงการตกลงเพิ่ม Corona Simulater

บทที่ 3

คำสั่งพื้นฐานของภาษา Lua

กฎการตั้งชื่อ

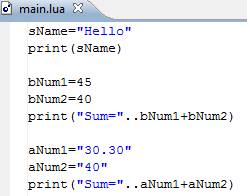
ชื่อ (ที่เรียกว่าตัวระบุ) หรือคำ ที่ใช้ใน Lua สามารถคำสั่ง ชนิดของตัวอักษร ตัวเลข ตัวบ่งชี้ตัวแปรชื่อและเขตข้อมูลตาราง แต่มีคำสงวนต่อไปนี้ไม่สามารถนำมาตั้งเป็นชื่อได้

and break do else elseif end false

for function if in local nil not

or repeat return then true until while

3.1 การประกาศตัวแปร print และ ผลรวมเลข



รูปที่ 3.1 แสดงการตัวอย่างประกาศตัวแปร การใส่ข้อความ การรวมเลข

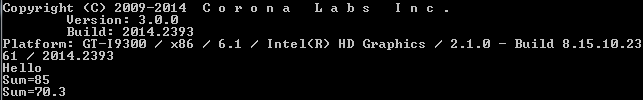
1.กำหนดตัวแปร sName = Sawasdee

2.ใช้คำสั่ง print (ตัวแปลที่กำหนด) แสดงข้อความเป็นตัวอักษร

3.กำหนดค่าตัวแปร bNum1= 45 และ bNum2= 40 \* aNam1, bNam2 = bNam2, aNam1

4 ใช้คำสั่ง print (“Sum (ผลรวม)”..bNum1+Num2)

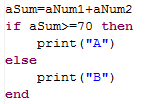
5. การกำหนดตัวแปลใช้เลขเป็นทดนิยมสามารถกำหนดได้เช่นกัน



รูปที่ 3.2 ผลจากการรันตัวอย่างประกาศตัวแปร การใส่ข้อความ การรวมเลข

3.2 การสร้างเงื่อนไข if else

การใช้ทางเลือกสองทางเลือก แสดงค่าในทางใดทางหนึ่งตามเงื่อนไขที่กำหนด



รูปที่ 3.3 แสดงตัวอย่าง if else

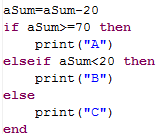
1.กำหนดโจทย์เงื่อนไข

2. if เงื่อนไข then   
ถ้าค่าของ aSum มากกว่า หรือเท่ากับ 70ให้แสดงค่า A  
else  
แสดงค่า B

รูปที่ 3.4 ผลจากการรันตัวอย่าง if else

ผลลัพธ์ออกเป็นค่า A แสดงว่าค่า aNum1+bNum2 มากกว่า 70

3.3 การสร้างเงื่อนไขงานที่มีมากกว่า 1 เงื่อนไข เราอาจใช้ if .. elseif .. else .. end



รูปที่ 3.5 แสดงตัวอย่าง if .. elseif .. else .. end

1.กำหนดโจทย์เงื่อนไข

2. if เงื่อนไข then

ค่าของ aSum มากกว่า หรือ เท่ากับ 70 ให้แสดงค่า A

elseif

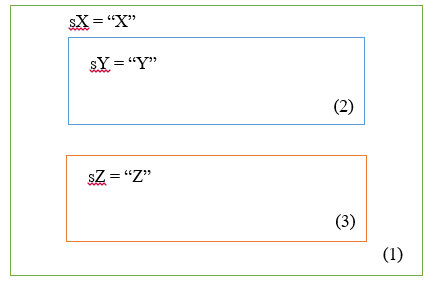
ค่าของ aSum น้อยกว่า 20 ให้แสดงค่า B

else

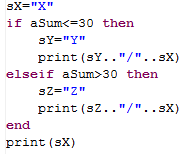
แต่ถ้าไม่เข้าเงื่อนไขไหนเลยให้แสดงค่า C

รูปที่ 3.6 ผลจากการรันตัวอย่าง if .. elseif .. else .. end

ผลลัพธ์ออกเป็นค่า C แสดงว่าค่า aSum-20 อยู่ระหว่างค่า 20 กับ 70

3.3.1 ตัวอย่าง if .. elseif .. else .. end

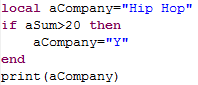
รูปที่ 3.7 แสดงตัวอย่าง if .. elseif .. else .. end

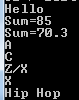
จากรูปจะเห็นว่า sX ถูกประกาศไว้ข้างนอก นั้นแสดงว่า sX ใช้ได้ทั้งหน้าคือ (1),(2),(3) ส่วน sY สามารถใช้ได้เฉพาะใน (2) เท่านั้น และsZ ก็สามารถใช้ได้เฉพาะใน (3) เท่านั้น   
เขียนได้ดั้งนี้

รูปที่ 3.8 การเขียนเงื่อนไข if .. elseif .. else .. end

รูปที่ 3.9 แสดงผลจากการรันผลลัพธ์ออกเป็นค่า Z/X แสดงว่าค่า X มากกว่า 30

3.4 การสร้างเงื่อนไข local if .. then

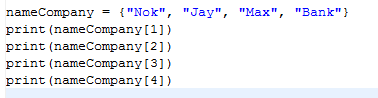
ในการใช้งานตัวแปรนั้นใน Lua จะมีทั้งตัวแปร global ( ตัวแปรปกติ ที่เราไม่ต้องประกาศอะไร) และตัวแปร local ที่เวลาจะใช้งานเราต้องประกาศชนิดตัวแปร local ให้มันก่อน

รูปที่ 3.10 แสดงตัวอย่าง local if .. then

รูปที่ 3.11 แสดงผลจากการรันตัวอย่าง local if .. then

ผลลัพธ์ออกเป็น Hip Hop แสดงว่าค่า aSum ต่ำกว่า 20

3.4.1 การสร้างเงื่อนไข Table fields ที่สามารถเก็บค่าได้มากกว่าหนึ่งค่า

เรื่องของ Table fields (properties) เป็นตัวแปรที่เก็บข้อมูลในแนวของ array คือตัวแปรที่สามารถเก็บค่าได้มากกว่าหนึ่งค่า เอาค่า 4 ค่าเก็บในตัวแปร nameCompany

รูปที่ 3.12 แสดงตัวอย่าง Table fields



รูปที่ 3.13 แสดงผลจากการรันตัวอย่าง Table fields

3.5 การสร้างเงื่อนไข Loop

เรื่องของการวน Loop (การวนซ้ำ) คือ โปรแกรมจะทำงานวนซ้ำไปเรื่อยๆตามเงื่อนไขที่เรากำหนดจนกว่าจะไม่เข้าเงื่อนไขแล้วถึงจึงจะออกจากระบบ การวนLoop

3.4.1 เงื่อนไข for..do

for ตัวแปรเริ่ม=? , ค่าที่จะทำการหยุด, ทำอะไรกับตัวแปรเริ่ม do

รูปที่ 3.14 แสดงตัวอย่างเงื่อนไข for..do

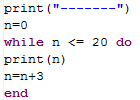
1.For ตัวแปร i เริ่มต้นที่ 5, สิ้นสุด 15 , โดยเพิ่มทีละ 2

2.แสดงค่าตัวแปร i

รูปที่ 3.15 แสดงผลจากการรันตัวอย่างเงื่อนไข for..do

3.4.2 เงื่อนไข while..do

while เงื่อนไข do



รูปที่ 3.16 แสดงตัวอย่าง เงื่อนไข while..do

1. whileให้ตัวแปร n เริ่มที่ 0 ถ้าค่าในตัวแปร n ยังน้อยกว่าหรือเท่ากับ 20 ให้doใน loop

2.แสดงค่าที่อยู่ในตัวแปร n ออกมา

3.เพิ่มค่าในตัวแปร n ทีละ 3 ทีละ 1 loop

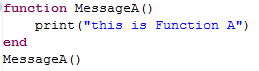


รูปที่ 3.17 แสดงผลจากการรันตัวอย่าง เงื่อนไข while..do

3.5 การสร้างเงื่อนไข function

เรื่องของ function ในกรณีที่ไม่อยากเขียนโค้ดซ้ำกันหลายรอบ ก็นำโค้ดนั้นมาทำเป็น function โดยจะต้องมีการสร้าง function และ การเรียกใช้ function ที่เราสร้างขึ้น ซึ่งลักษณะของ function ใน Lau มี

3.5.1 function ที่ไม่รับค่าและไม่ส่งค่าออก



-

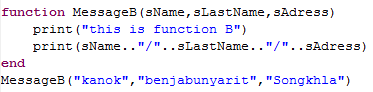
รูปที่ 3.18 แสดงตัวอย่าง function ที่ไม่รับค่าและไม่ส่งค่าออก

function ที่ไม่รับค่าและไม่ส่งค่าออก

1. สร้าง function ชื่อ MessageA ที่ไม่มีการรับค่า( ) และไม่มีการส่งค่าออกมา(return)

2. เรียกใช้ function MessageA ที่สร้างขึ้น

รูปที่ 3.19 แสดงผลจากการรันตัวอย่าง function ที่ไม่รับค่าและไม่ส่งค่าออก

3.5.2 function ที่รับค่าและไม่ส่งค่าออก

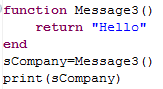
รูปที่ 3.20 แสดงตัวอย่าง function ที่รับค่าและไม่ส่งค่าออก

1. สร้าง function ชื่อ MessageB มีการรับค่า 3 ค่า ( .. , .. , .. )

2. และไม่มีการส่งค่าออกมา (return)

3. เรียกใช้ function MessageB ที่สร้างขึ้น

รูปที่ 3.21 แสดงผลจากการรันตัวอย่าง function ที่รับค่าและไม่ส่งค่าออก

3.5.3 function ที่ไม่รับค่าและส่งค่าออก

รูปที่ 3.22 แสดงตัวอย่าง function ที่ไม่รับค่าและส่งค่าออก

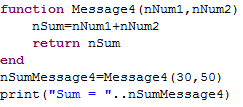
1. สร้าง function ชื่อ Message3 ที่ไม่มีการรับค่า ( )

2.และมีการส่งค่าออกมา (return)

3. เรียกใช้ function Message3 ที่สร้างขึ้น



รูปที่ 3.23 แสดงผลจากการรันตัวอย่าง function ที่ไม่รับค่าและส่งค่าออก

3.5.4 function ที่รับค่าและส่งค่าออก

รูปที่ 3.24 แสดงตัวอย่าง function ที่รับค่าและส่งค่าออก

1. สร้าง function ชื่อ Message 4 มีการรับค่า 2 ค่า ( .. , .. )

2. และมีการส่งค่าออกมา (return)

3. เรียกใช้ function Message4 ที่สร้างขึ้น



รูปที่ 3.25 แสดงผลจากการรันตัวอย่าง function ที่รับค่าและส่งค่าออก

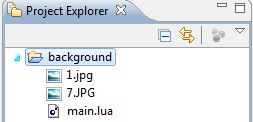
บทที่ 4

การใช้คำสั่ง Corona และ Lua เบื้องต้น

4.1 ขั้นตอนการใส่รูปภาพในโฟลเดอร์

เป็นขั้นตอนแสดงการใส่รูปพื้นหลังของแอพพลิเคชั่น

1 ให้เลือกรูปภาพที่ต้องการนำใส่ลงในโฟลเดอร์ ตั้งชื่อว่า “background “



รูปที่4.1 แสดงการนำรูปลงใส่โฟลเดอร์

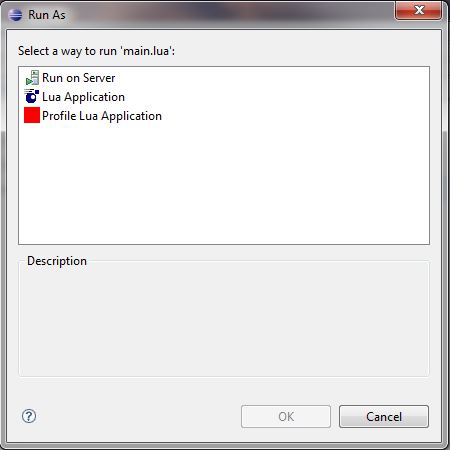
4.2 ขั้นตอนการใส่ข้อความ

1. ให้ทำการพิมพ์ code display.newtext ซึ่งจะมีการกำหนดข้อความ แกน X แกน Y Font และ ขนาดของตัวอักษร



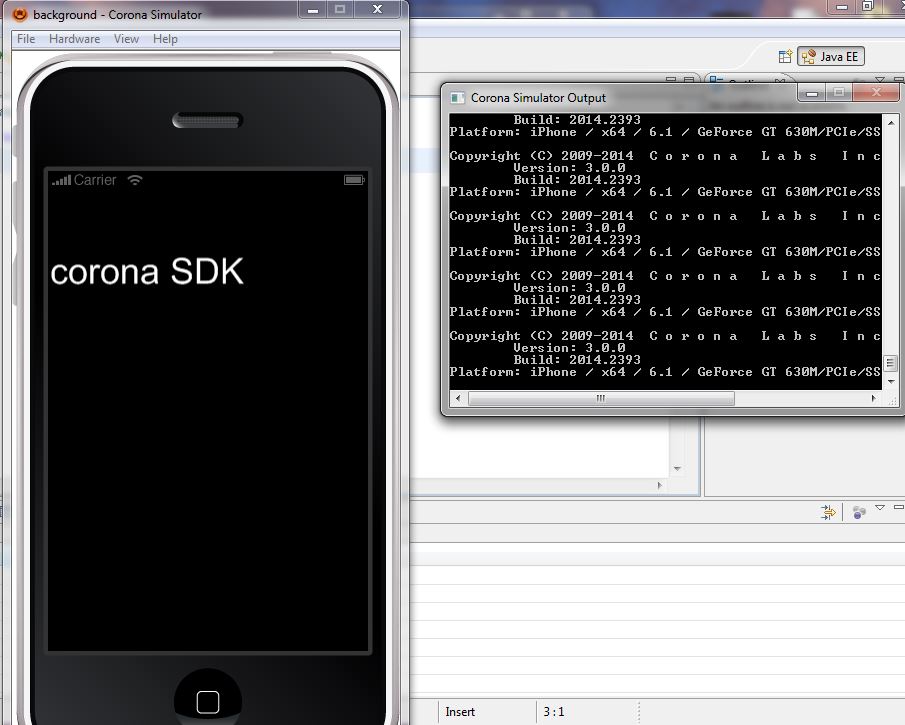
รูปที่4.2 แสดงการข้อความ

2. ให้กด Lua Application แล้วกด Ok



รูปที่4.3 แสดงการเลือกใช้ Lua Application

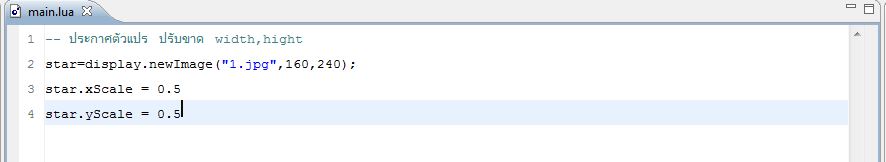
3 เมื่อกด run จะได้ข้อความตาม code ที่พิมพ์ ดังนี้



รูปที่4.4 แสดงผลลัพธ์การใส่ข้อความ

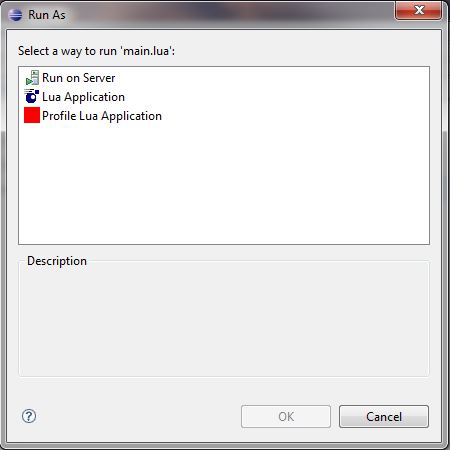
4.3 ขั้นตอนการใส่พื้นหลัง

1 ให้พิมพ์ข้อความดังรูป โดยจะมีการประกาศตัวแปร ขนาด Width, Height และองศาภาพ



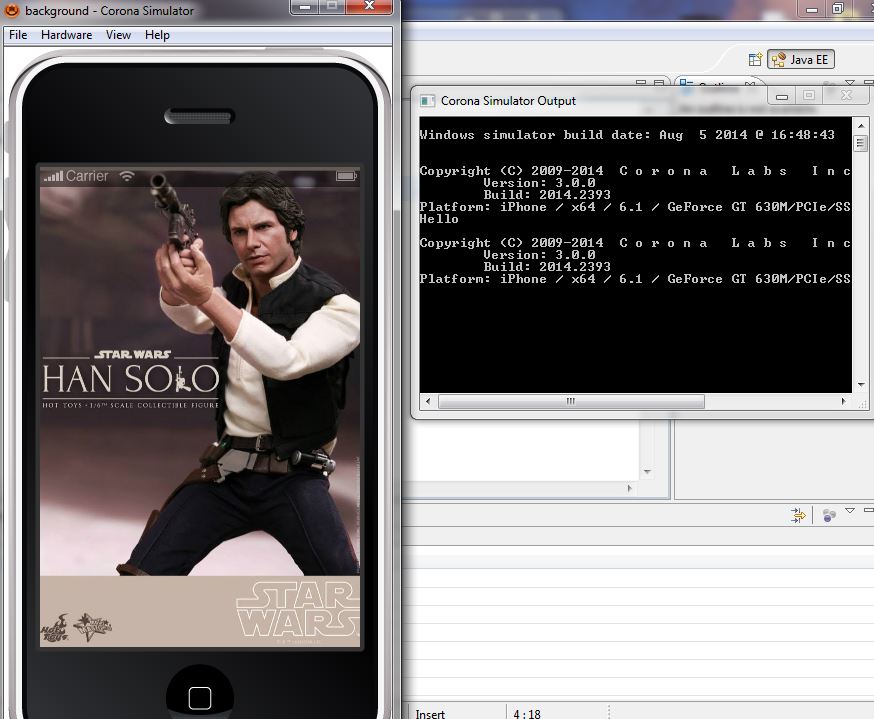
รูปที่ 4.5 แสดงการกำหนดขนาดพื้นหลัง

2.ให้กด Lua Application แล้วกด Ok



รูปที่4.5 แสดงการเลือกใช้ Lua Application

3 .เมื่อกด run จะได้ผลลัพธ์ตาม code ที่พิมพ์ ดังรูป



รูปที่4.6 แสดงผลลัพธ์การใส่พื้นหลัง

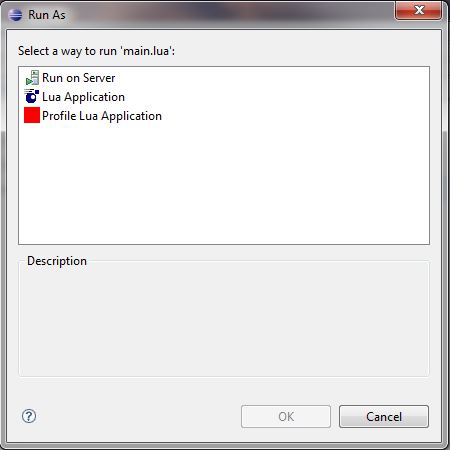
4.4 ขั้นตอนการให้รูปเคลื่อนไหว

1 ให้พิมพ์ code ตามข้อความการกำหนดรูปและขนาดของภาพ



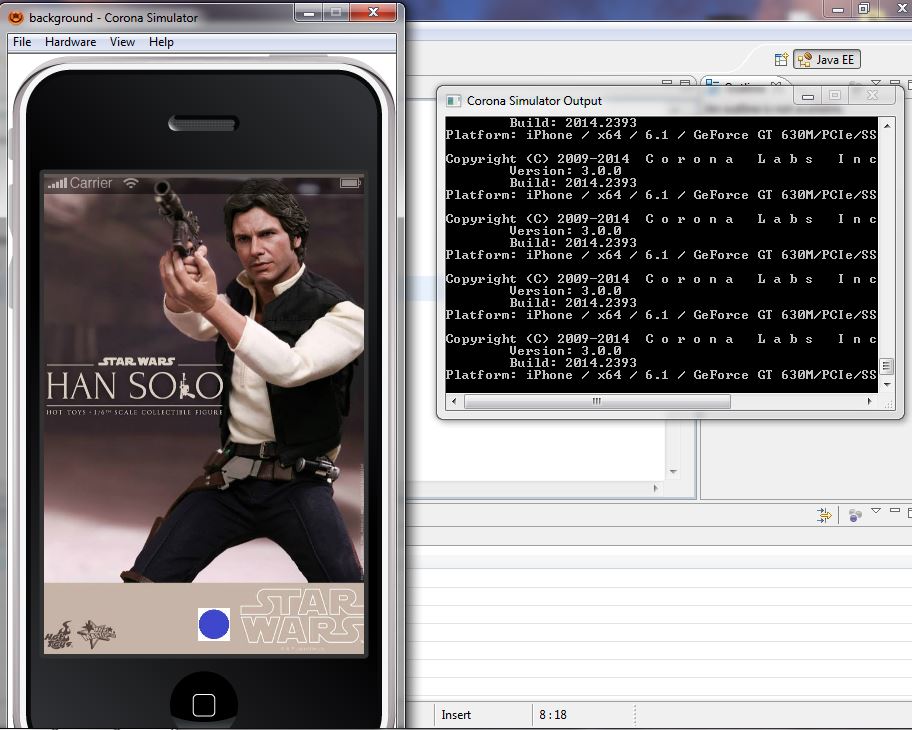
รูปที่4.7 แสดงการกำหนดรูปและขนาดของภาพ

2 ให้กด Lua Application แล้วกด Ok



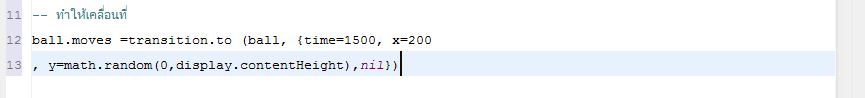
รูปที่4.8 แสดงการเลือกใช้ Lua Application

3 เมื่อกด run จะได้ผลลัพธ์ตาม code ที่พิมพ์ ดังรูป (เป็นลูกบอลสีฟ้า)



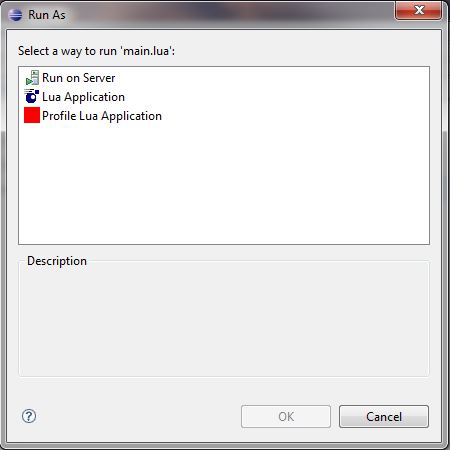
รูปที่4.9 แสดงผลลัพธ์ใส่รูปลูกบอล

4. ให้พิมพ์ code ดังรูป โดยจะมีการกำหนด เวลาในการเคลื่อนที่ แกน X แกน Y ในการเลื่อนตำแหน่ง



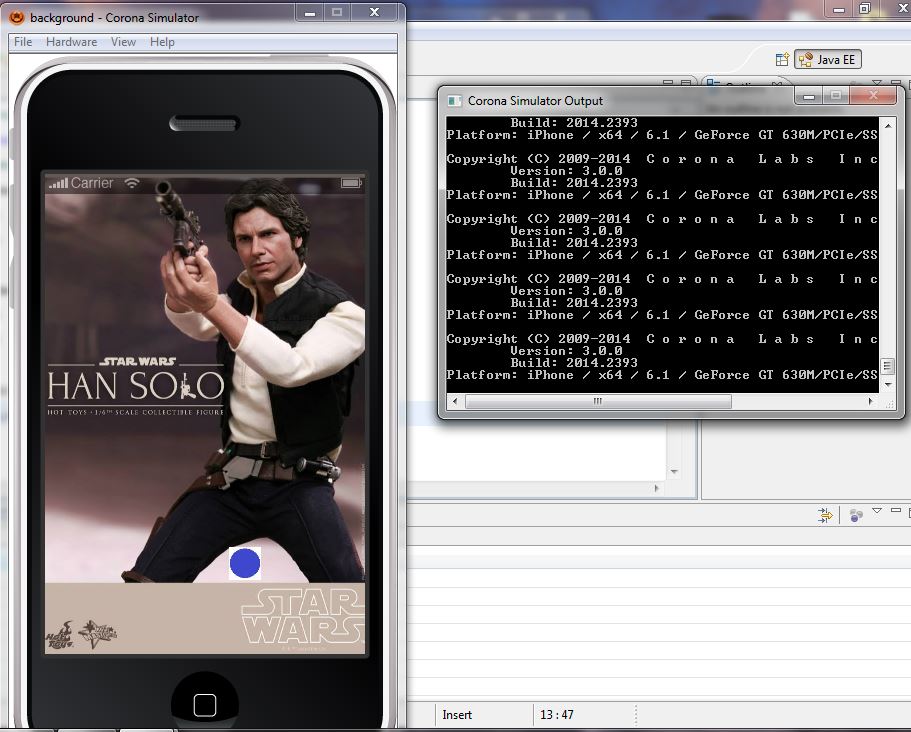
รูปที่4.10 แสดงการกำหนดเวลาในการเคลื่อนที่ แกน X แกน Y ในการเลื่อนตำแหน่ง

5.ให้กด Lua Application แล้วกด Ok



รูปที่4.11 แสดงการเลือกใช้ Lua Application

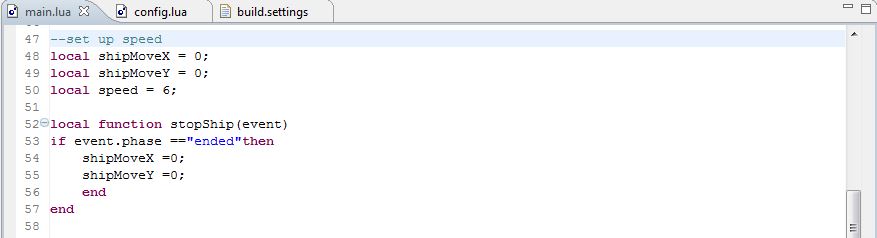
6.เมื่อกด run จะได้ผลลัพธ์ตาม code ที่พิมพ์ ดังรูป (เป็นลูกบอลสีฟ้าจะเคลื่อนขึ้น)



รูปที่4.12 แสดงผลลัพธ์การให้รูปเคลื่อนไหว

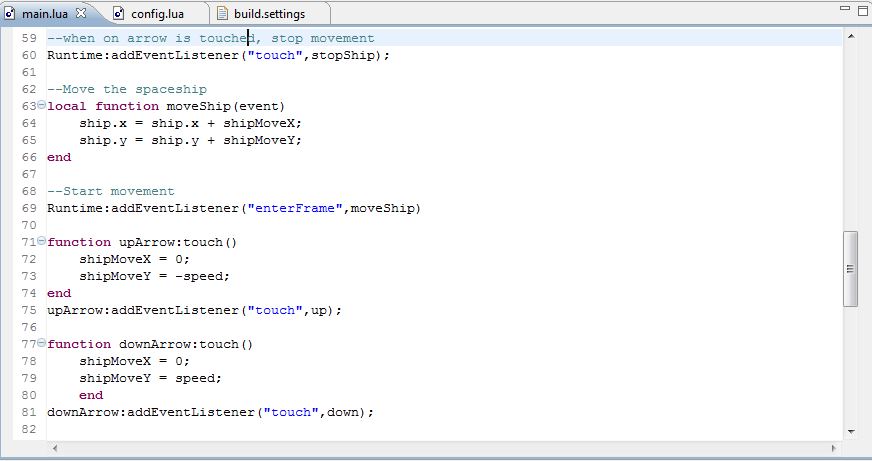
4.5 ขั้นตอนการทำให้ลูกศรคลิก และเลื่อนรูปภาพตามลูกศรที่คลิก

1 ให้พิมพ์ code โดยการ set up speed จะเป็นการกำหนดความเร็วของการเคลื่อนที่เครื่องบิน และ การกำหนดให้เครื่องบินหยุดนิ่ง

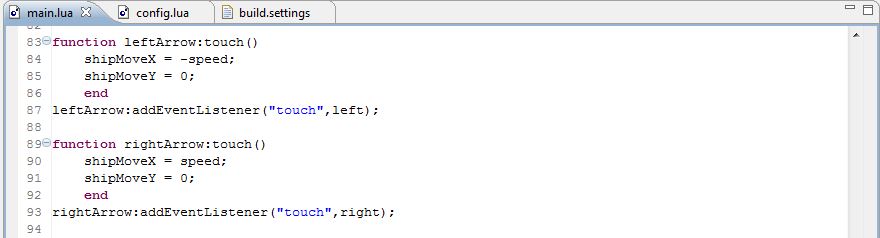


รูปที่4.13 แสดงการกำหนดความเร็วของการเคลื่อนที่เครื่องบิน และ การกำหนดให้เครื่องบินหยุดนิ่ง

2. การกำหนดการคลิกจากลูกศรให้เครื่องบินไปตามที่คลิก

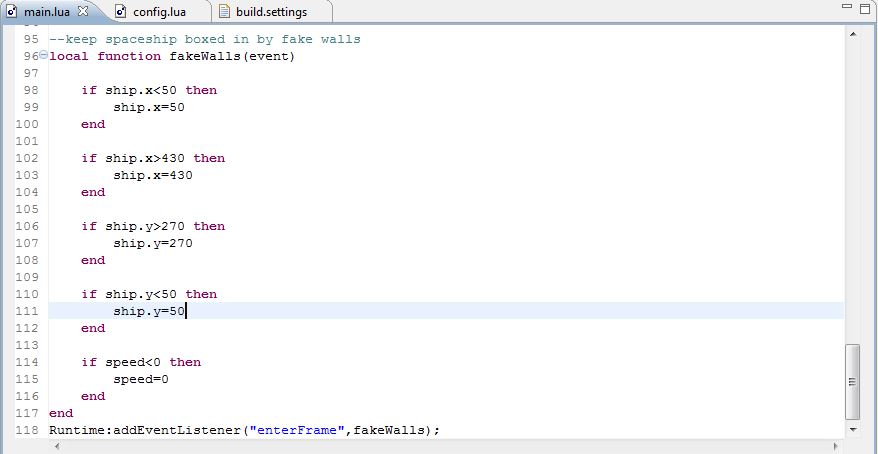


รูปที่4.14 แสดงการกำหนดการคลิกจากลูกศรให้เครื่องบินไปตามที่คลิก



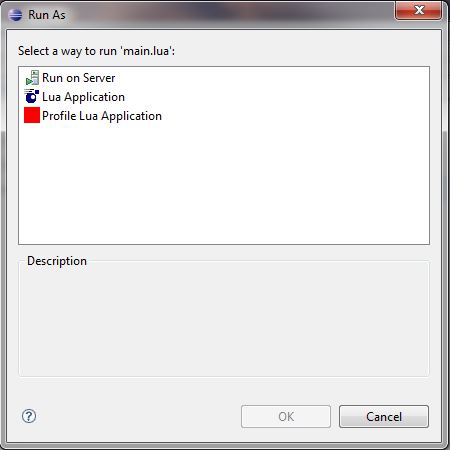
รูปที่4.14 แสดงการกำหนดการคลิกจากลูกศรให้เครื่องบินไปตามที่คลิก

3 การกำหนดการติดหน้าจอเครื่องบิน



รูปที่4.15 แสดงการกำหนดการติดหน้าจอเครื่องบิน

4 ให้กด Lua Application > ok



รูปที่4.16 แสดงแสดงการเลือกใช้ Lua Application

5 การแสดงผลของ code



รูปที่4.17 แสดงการแสดงผลของ code

6 ผลลัพธ์หลังกดลูกศรทางขวา



รูปที่4.18 แสดงผลลัพธ์หลังกดลูกศรทางขวา

บรรณานุกรม