

บทที่ 5

ตัวแปร

5.1 รู้จักกับตัวแปรในภาษาโลโก

พิจารณาคำสั่งที่สั่งให้เต่าทำงานจะพบว่าคำสั่งส่วนใหญ่จะตามด้วยข้อมูลจำนวน เช่น คำสั่ง FD 20 จำนวน 20 คือค่าที่ระบุว่าจะให้เต่าเดินหน้ากี่หน่วย คำสั่ง RT 30 จำนวน 30 คือค่าที่ระบุว่าจะให้เต่าหันขวากี่องศา ข้อมูลจำนวนเหล่านี้จะเป็นค่าคงที่ ซึ่งหากต้องการใช้คำสั่งเดิมแต่เปลี่ยนค่าจะไม่สามารถเปลี่ยนแปลงในกระบวนการความได้ โลโกได้อำนวยความสะดวกในการเปลี่ยนค่าเหล่านี้ โดยให้กำหนดเป็นตัวแปร (variable) และกำหนดค่าตัวแปรในภายหลัง

ตัวแปรในภาษาโลโกสามารถกำหนดชนิดได้หลายชนิด เช่น ชนิดตัวเลข (numeric) ชนิดสายอักขระ (string) ตัวแปรที่จะกล่าวถึงในบทนี้จะเป็นชนิดตัวเลข

ในการกำหนดตัวแปรของภาษาโลโกมี 2 วิธี วิธีแรกเป็นการใช้คำสั่ง MAKE และวิธีใช้กระบวนการที่มีการกำหนดให้ใส่ค่าตัวแปร ซึ่งจะมีรูปแบบในลักษณะเช่นเดียวกับการใส่ค่าคงที่ใน คำสั่ง FORWARD BACK RIGHT เป็นต้น

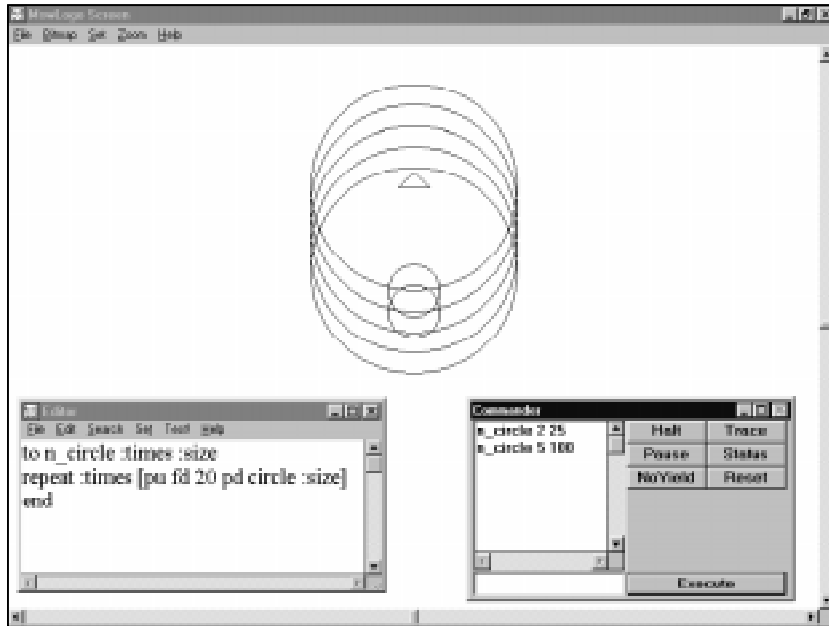
การที่เราสามารถกำหนดค่าตัวแปรให้กับกระบวนการ จะสามารถใช้กระบวนการทำงานได้อย่างมีประสิทธิภาพมากขึ้น เช่น วาดรูปได้หลายขนาดตามที่ต้องการ วาดรูปได้หลากหลายแบบจากค่าตัวแปรที่กำหนด เป็นต้น

5.2 รูปแบบกระบวนการที่มีการกำหนดค่าตัวแปร

รูปแบบกระบวนการที่มีการกำหนดค่าตัวแปรเป็นดังนี้

TO ชื่อกระบวนการ :ตัวแปร1 :ตัวแปร2 : ...

ตัวอย่างที่ 5.1 กระบวนการความลั่งเต่าวาดรูปวงกลมที่ระบุค่าเป็นตัวแปร



รูปที่ 5.1 กระบวนการความ `n_circle :time :size` และผลการใช้กระบวนการความ

จากรูป ในกระบวนการความ `n_circle` ประกอบด้วยตัวแปร 2 ตัว คือ `:times` ซึ่งเป็นตัวกำหนดค่าจำนวนวงกลม และ `:size` ซึ่งเป็นตัวกำหนดค่าขนาดของวงกลม จึงเป็นกระบวนการที่อนุญาตให้เรากำหนดจำนวนวงกลม และขนาดได้ในภายหลัง

เมื่อสั่งให้เต่าทำงานด้วยคำสั่ง `n_circle 2 25` เต่าจะวาดรูปวงกลมที่มีรัศมี 25 หน่วย จำนวน 2 วงและคำสั่ง `n_circle 5 100` เต่าจะวาดรูปวงกลมที่มีรัศมี 100 หน่วย จำนวน 5 วง ดังผลลัพธ์ที่ได้ในส่วนแสดงผลกราฟิก

5.3 การใช้ตัวแปรเปลี่ยนขนาดรูป

จากบทที่ผ่านมาถ้าต้องการสร้างรูปสี่เหลี่ยม 5 รูป ให้มีขนาดต่างกันซ้อนกัน จะต้องใช้กระบวนการความ 5 กระบวนการที่กำหนดค่าขนาดของด้านต่างกัน โดยในคำสั่ง FORWARD จะต้องตามด้วยตัวเลขที่เป็นค่าที่แตกต่างกันทั้ง 5 กระบวนการ

แต่ถ้าสร้างกระบวนการที่กำหนดตัวแปรเพื่อรับค่าขนาดของด้าน จะสามารถใช้กระบวนการนี้กระบวนการเดียว สั่งเต่าวาดรูปสี่เหลี่ยมตามขนาดที่ต้องการโดยกำหนดค่าความยาวของด้านให้กับตัวแปร

ตารางที่ 5.1 เปรียบเทียบกระบวนการ SQUARE ซึ่งใช้สั่งเต่าวาดรูปสี่เหลี่ยมที่มีการกำหนดขนาดคงที่ กับ กระบวนการ S_SQUARE ซึ่งใช้สั่งเต่าวาดรูปสี่เหลี่ยมเหมือนกันแต่กำหนดตัวแปรที่สามารถรับค่าขนาดของด้านได้ ดังนี้

กระบวนการ SQUARE	กระบวนการ S_SQUARE
TO SQUARE FORWARD 50 RIGHT 90 FORWARD 50 RIGHT 90 FORWARD 50 RIGHT 90 FORWARD 50 RIGHT 90 END	TO S_SQUARE :SIZE FORWARD :SIZE RIGHT 90 FORWARD :SIZE RIGHT 90 FORWARD :SIZE RIGHT 90 FORWARD :SIZE RIGHT 90 END
หรือ TO SQUARE REPEAT 4 [FORWARD 50 RIGHT 90] END	หรือ TO S_SQUARE :SIZE REPEAT 4 [FORWARD :SIZE RIGHT 90] END

การใช้กระบวนการ S_SQUARE สามารถกำหนดจำนวนใด ๆ แทนตัวแปร :SIZE ทำให้สามารถสั่งเต่าวาดรูปสี่เหลี่ยมขนาดใดก็ได้ โดยการพิมพ์คำสั่ง S_SQUARE ตามด้วยจำนวน เช่น S_SQUARE 100 เป็นการสั่งเต่าวาดรูปสี่เหลี่ยมที่มีพื้นที่ขนาด 100 X 100 ตารางหน่วย

หมายเหตุ ข้อควรระวังในการกำหนดตัวแปร

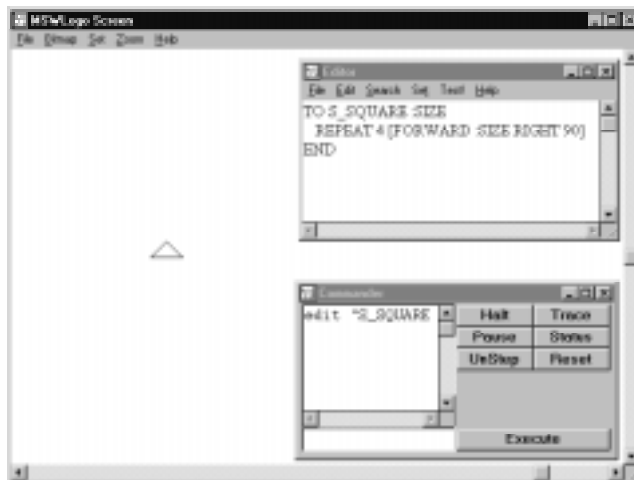
- 1) เมื่อพิมพ์ :SIZE ต้องไม่เว้นวรรคระหว่าง : และ SIZE
- 2) ไม่ต้องพิมพ์ : เมื่อใส่ค่าของตัวแปร SIZE เช่น ถ้าพิมพ์ S_SQUARE :100 เต่าจะแสดงข้อความว่า 100 has no value หมายความว่า ตัวแปรชื่อ 100 ไม่มีค่าข้อมูล เนื่องจากเมื่อพิมพ์ :100 เต่าจะเห็นว่า 100 คือ ชื่อตัวแปร และค้นหาค่าข้อมูลที่เก็บในตัวแปรชื่อ 100 มาใช้งาน แต่หาไม่พบ จึงแสดงข้อความข้างต้น

กิจกรรมที่ 5.1

กระบวนการวาดรูปสี่เหลี่ยมขนาดต่างๆ

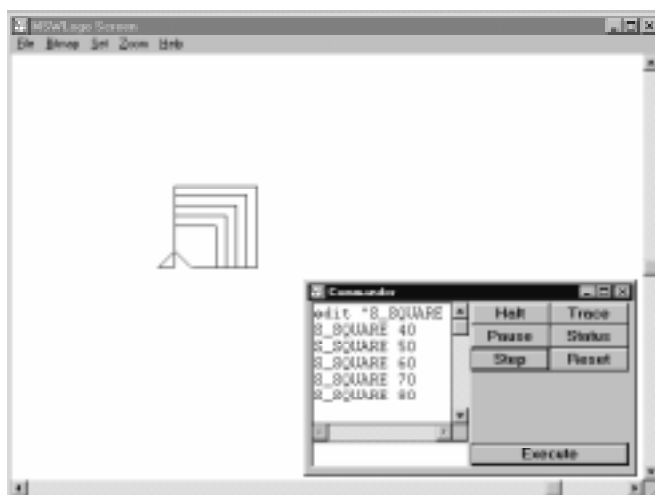
ต้องการวาดรูปสี่เหลี่ยมขนาดต่าง ๆ ทำได้ดังนี้

- 1) สร้างกระบวนการ S_SQUARE ดังรูปที่ 5.2



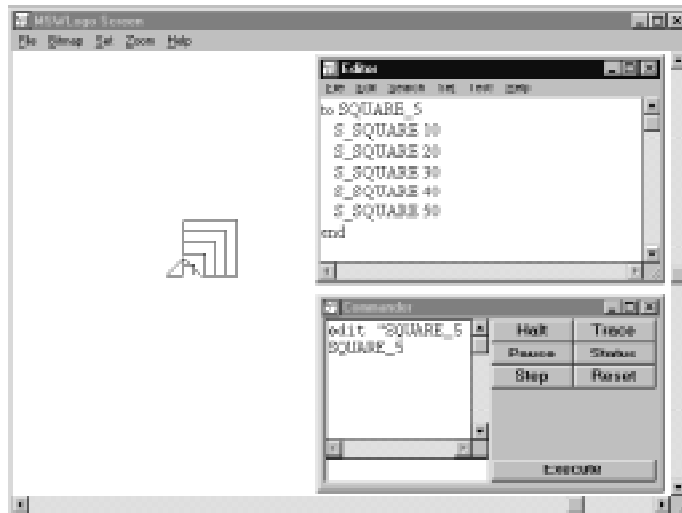
รูปที่ 5.2 กระบวนการ S_SQUARE

- 2) ในช่องป้อนคำสั่ง ให้สั่งเต่าวาดรูปสี่เหลี่ยมห้ารูปที่มีขนาดต่าง ๆ ตามลำดับคำสั่งในรูปที่ 5.3 แล้วเปรียบเทียบผลที่ได้บนจอภาพกับรูปที่ 5.3



รูปที่ 5.3 ผลที่ได้เมื่อสั่งเต่าวาดรูปสี่เหลี่ยมห้ารูป

- 3) สร้างกระบวนการความชื่อ SQUARE_5 แล้วใช้กระบวนการนี้สั่งเต่าวาดรูปสี่เหลี่ยมจำนวน 5 รูป ดังรูปที่ 5.4



รูปที่ 5.4 ผลที่ได้เมื่อสั่งเต่าวาดรูปสี่เหลี่ยม 5 รูป ด้วยคำสั่ง SQUARE_5

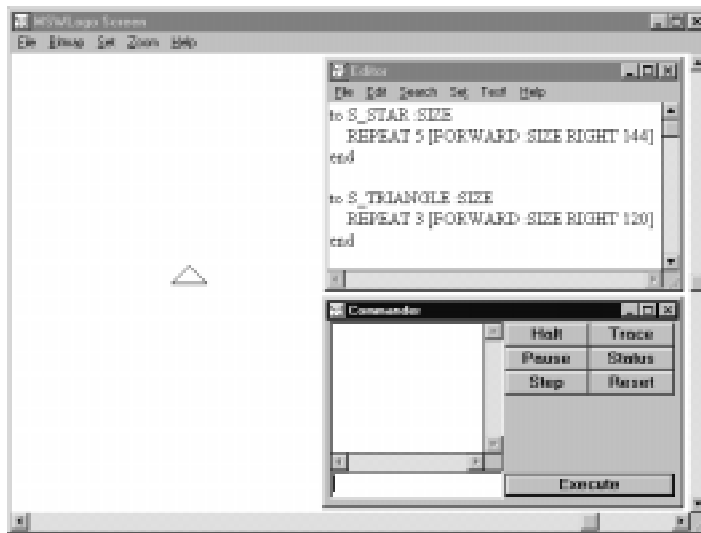
- 4) บันทึกกระบวนการทั้งหมดลงในแฟ้มชื่อ pratic_5
5) อภิปรายกลุ่มว่ากิจกรรมในข้อ 3) ทำเพื่ออะไร

กิจกรรมที่ 5.2

ฝึกสร้างกระบวนการวาดรูปขนาดต่างๆ

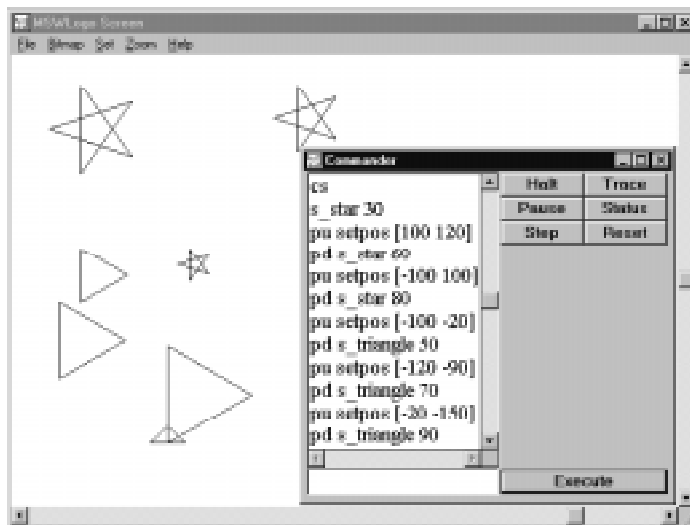
จากบทที่ 4 เราเคยสร้างกระบวนการ STAR และกระบวนการ TRIANGLE เพื่อสั่งเต่าวาดรูปดาว และสามเหลี่ยมที่มีขนาดคงที่ ในกิจกรรมนี้จะสร้างกระบวนการ S_STAR และกระบวนการ S_TRIANGLE วาดรูปดาวและสามเหลี่ยมที่มีขนาดต่าง ๆ ดังนี้

- 1) สร้างกระบวนการ S_STAR และกระบวนการ S_TRIANGLE ดังรูปที่ 5.5



รูปที่ 5.5 กระบวนการวาด S_STAR และกระบวนการวาด S_TRIANGLE

2) กระบวนการวาด S_STAR และกระบวนการวาด S_TRIANGLE สั่งเต่าวาดรูป ดังรูปที่ 5.6



รูปที่ 5.6 รูปดาวและสามเหลี่ยมที่มีขนาดต่าง ๆ กัน

หมายเหตุ คำสั่ง SETPOS [X Y] เป็นคำสั่งพื้นฐานสำหรับใช้เคลื่อนย้ายเต่าไปยังตำแหน่งที่ต้องการ โดยค่า X และ Y เป็นค่าคู่ลำดับในแกน X และแกน Y
อย่าลืมว่า ตำแหน่งบ้านของเต่าอยู่ที่คู่ลำดับ X=0 และ Y=0

3) บันทึกกระบวนการทั้งหมดลงในแฟ้ม pratic_5

5.4 แนวคิดเกี่ยวกับการใช้ตัวแปรในภาษาโลโก

มีแนวคิดเกี่ยวกับตัวแปรที่น่าสนใจมากมาย เช่น

- 1) ตัวแปร หมายถึง ข้อมูลที่มีการกำหนดค่าและเปลี่ยนแปลงค่าด้วยคำสั่งภายในโปรแกรม ประกอบด้วย ชื่อตัวแปร ค่าตัวแปร และตำแหน่งที่เก็บค่าตัวแปรในหน่วยความจำ
- 2) เมื่อสร้างกระบวนการที่มีตัวแปร เช่น TO S_SQUARE :SIZE เต่าจะเก็บกระบวนการชื่อ S_SQUARE ในหน่วยความจำ และเตรียมเนื้อที่ว่างในหน่วยความจำสำหรับเก็บค่าของตัวแปรชื่อ SIZE ไว้
- 3) เมื่อสั่งเต่าให้ทำงานตามกระบวนการที่มีตัวแปร เช่น คำสั่ง S_SQUARE 20 เต่าจะรับรู้ตัวเลขจำนวน 20 คือค่าของตัวแปรชื่อ SIZE และนำไปเก็บในเนื้อที่ว่างในหน่วยความจำที่เตรียมไว้ เพื่อนำไปสร้างกระบวนการ S_SQUARE ต่อไป
- 4) ในการทำงานตามกระบวนการที่มีตัวแปร เช่น S_SQUARE :SIZE เครื่องหมาย : จะบอกเต่าให้หาค่าของตัวแปรชื่อ SIZE ในเนื้อที่ว่างในหน่วยความจำที่เตรียมไว้ และนำค่าของตัวแปรไปใช้เหมือนกับการส่งค่าตัวแปรให้กระบวนการ
- 5) ข้อแตกต่างระหว่างเครื่องหมาย : และ “ ในภาษาโลโกมีดังนี้คือ
 - 5.1) การกำหนดให้มีเครื่องหมาย : นำหน้าตัวแปรใด หมายถึง การอ้างถึงค่าของตัวแปร ที่จัดเก็บในตัวแปรชื่อนั้น เช่น :SIZE จะหมายถึงค่าของตัวแปรที่เก็บในตัวแปรชื่อ SIZE
 - 5.2) การกำหนดให้มีเครื่องหมาย “ หน้าตัวแปรใด หมายถึง การอ้างถึงชื่อของตัวแปรนั้น เช่น “SIZE จะหมายถึง ชื่อของตัวแปรที่เรียกว่า SIZE
 - 5.3) การกำหนดตัวแปรโดยไม่เครื่องหมาย : หรือ “ นำหน้า หมายถึง ตัวแปรนั้นเป็นคำสั่งหรือกระบวนการ เช่น SIZE ที่ไม่มีเครื่องหมาย : หรือ “ นำหน้า เต่าจะเข้าใจว่าเป็นคำสั่งหรือกระบวนการที่มีชื่อว่า SIZE

6) ตัวแปรที่ใช้เฉพาะในกระบวนการ และตัวแปรสาธารณะ

- 6.1) ตัวแปรที่กำหนดในกระบวนการที่มีตัวแปร จะเป็นตัวแปรที่ใช้เฉพาะในกระบวนการนั้น เช่น TO S_SQUARE :SIZE ตัวแปรชื่อ SIZE จะได้รับการกำหนดเป็นตัวแปรสำหรับใช้เฉพาะในกระบวนการ S_SQUARE เมื่อสั่ง S_SQUARE 20 เลขจำนวน 20 จะถูกเก็บในเนื้อที่ว่างในหน่วยความจำว่า SIZE ซึ่งในขณะที่กำลังทำงานจะเก็บค่าเพียง 1 ค่า และจะถูกนำมาใช้งานในกระบวนการ S_SQUARE จนกระทั่งกระบวนการ เมื่อจบแล้วเนื้อที่ว่างในหน่วยความจำในชื่อว่า SIZE ซึ่งมีค่า 20 จะถูกทำให้ว่างอีกครั้ง
- 6.2) ตัวแปรที่กำหนดด้วยคำสั่ง MAKE จะเป็นตัวแปรแบบสาธารณะที่ใช้ได้ทั่วไปในชิ้นงาน รูปแบบของคำสั่ง MAKE มีดังนี้

MAKE “ชื่อตัวแปร ค่าตัวแปร

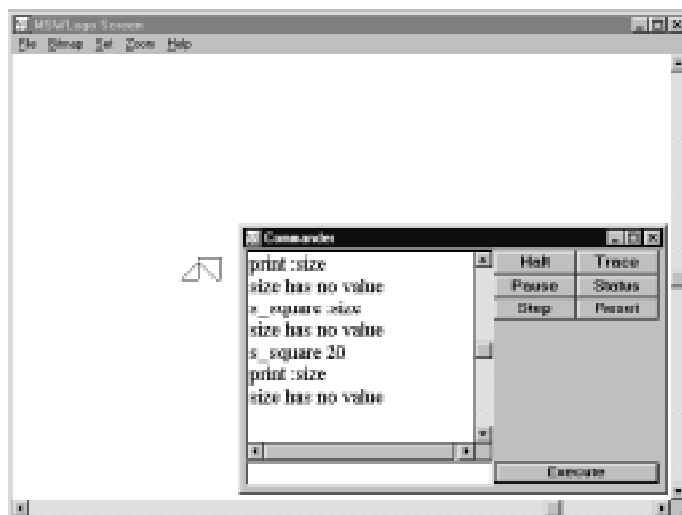
ผลของการกำหนดตัวแปรด้วยคำสั่ง MAKE โปรแกรมจะกำหนดเนื้อที่ว่างในหน่วยความจำสำหรับเก็บข้อมูลแบบสาธารณะมีชื่อตามชื่อตัวแปรที่กำหนด และนำค่าตัวแปรที่กำหนดให้ไปใส่ไว้ เช่น เมื่อกำหนดตัวแปรด้วยคำสั่ง MAKE “SIZE 50 เค้าจะกำหนดเนื้อที่ว่างในหน่วยความจำสำหรับเก็บข้อมูลแบบสาธารณะชื่อ SIZE และใส่ค่าตัวแปร 50 ลงไป

กิจกรรมที่ 5.3

ตัวแปรใช้เฉพาะในกระบวนความและตัวแปรสาธารณะ

กิจกรรมนี้จะช่วยสร้างความเข้าใจเกี่ยวกับตัวแปรใช้เฉพาะในกระบวนความและตัวแปรสาธารณะ ให้ทำตามขั้นตอนและสังเกตผลที่ได้ในแต่ละขั้นอย่างละเอียด ดังนี้

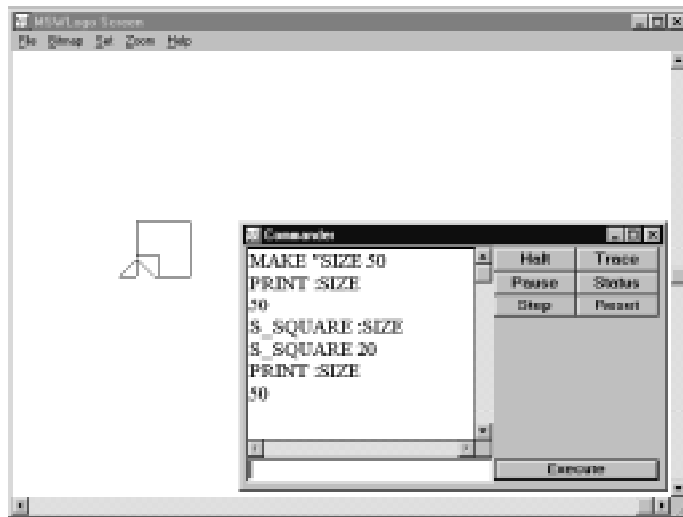
- 1) บรรจุเพิ่มสร้างกระบวนความ `pratic_5` ซึ่งมีกระบวนความ `S_SQUARE :SIZE` เข้ามาในหน่วยความจำ
- 2) สั่งเต้าให้ทำงานตามลำดับคำสั่งในรูปที่ 5.7



รูปที่ 5.7 แสดงค่าของตัวแปรที่ได้รับการกำหนดใช้เฉพาะในกระบวนความ

จากรูปจะพบว่า คำสั่ง `print :size` และ `s_square :size` ไม่มีค่าของตัวแปร `size` แต่ `s_square 20` รับค่า 20 ใส่ในตัวแปร `size` และใช้งานในกระบวนความจนจบ แล้วทำให้ตัวแปร `size` วางอีกครั้งหนึ่ง ดังนั้นเมื่อสั่ง `print :size` จึงไม่มีค่า `size`

- 3) สั่งเต้าให้ทำงานตามลำดับคำสั่งในรูปที่ 5.8



รูปที่ 5.8 แสดงค่าของตัวแปรที่ถูกกำหนดให้เป็นตัวแปรสาธารณะ

จากรูปคำสั่ง MAKE "SIZE 50 ทำให้เกิดตัวแปรขึ้นตัวหนึ่งชื่อว่า SIZE เป็นตัวแปรสาธารณะ เก็บค่า 50 ในขณะที่กระบวนการ S_SQUARE :SIZE มีตัวแปรชื่อ SIZE อีกตัวหนึ่งเป็นตัวแปรใช้เฉพาะกระบวนการ S_SQUARE ซึ่งมีค่าว่างเปล่าอยู่ คำสั่ง PRINT :SIZE จะนำค่า 50 ซึ่งเป็นค่าของตัวแปรสาธารณะชื่อ SIZE มาแสดง

คำสั่ง S_SQUARE :SIZE จะนำค่า 50 จากตัวแปรสาธารณะชื่อ SIZE มากำหนดให้เป็นค่าของตัวแปรใช้เฉพาะกระบวนการชื่อ SIZE ของ S_SQUARE ฉะนั้นคำสั่งนี้คือการสั่ง S_SQUARE 50 ผลที่ได้คือรูปสี่เหลี่ยมที่มีพื้นที่ขนาด 50 X 50 ตารางหน่วย

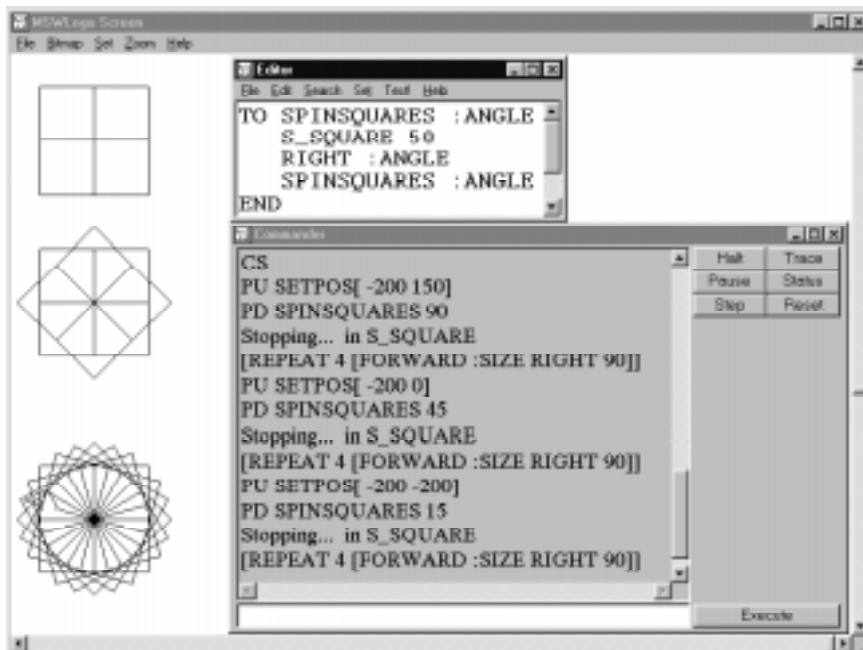
คำสั่ง S_SQUARE 20 จะนำค่า 20 ซึ่งเป็นค่าของตัวแปรใช้เฉพาะกระบวนการชื่อ SIZE มาวาดรูปสี่เหลี่ยมที่มีพื้นที่ขนาด 20 X 20 ตารางหน่วย โดยไม่เกี่ยวข้องกับค่าของตัวแปรสาธารณะชื่อ SIZE ดังนั้นเมื่อสั่ง PRINT :SIZE จึงได้ค่า 50

- 4) เหตุผลที่ตัวไม่มีค่าของตัวแปรชื่อ SIZE ในรูปที่ 5.7 แต่กลับมีค่าของตัวแปรชื่อ SIZE ทุกครั้งในรูปที่ 5.8 เพราะ

.....

5.5 การใช้ตัวแปรเปลี่ยนแปลงรูป

การเรียกกระบวนการที่มีตัวแปรโดยกำหนดค่าของตัวแปรต่างกัน อาจทำให้รูปที่วาดมีลักษณะแตกต่างกันได้ เช่น กระบวนการ SPINSQUARES :ANGLE มีตัวแปร ANGLE สำหรับรับข้อมูลที่ป้อนเข้ามาเป็นมุมต่าง ๆ เมื่อมีการใส่ค่าของมุมเป็น 90 45 และ 15 จะทำให้ได้วาดรูปที่มีรูปร่างต่างกัน ดังรูปที่ 5.9



รูปที่ 5.9 กระบวนการ SPINSQUARES : ANGLE และผลที่ได้เมื่อเปลี่ยนค่ามุม

กระบวนการ SPINSQUARES : ANGLE ใช้หลักการการซ้ำ เมื่อเต่าทำตามคำสั่งในกระบวนการจนถึงบรรทัดที่ 4 เต่าจะไปทำตามคำสั่งในกระบวนการ SPINSQUARES : ANGLE อีกครั้ง ซึ่งเป็นการเรียกตัวเองกลับมาทำงาน และจะวนทำตามคำสั่งในกระบวนการนี้เรื่อย ๆ ไป ถ้าต้องการให้หยุดต้องคลิกปุ่ม Halt เพื่อยกเลิกการทำงานของเต่า

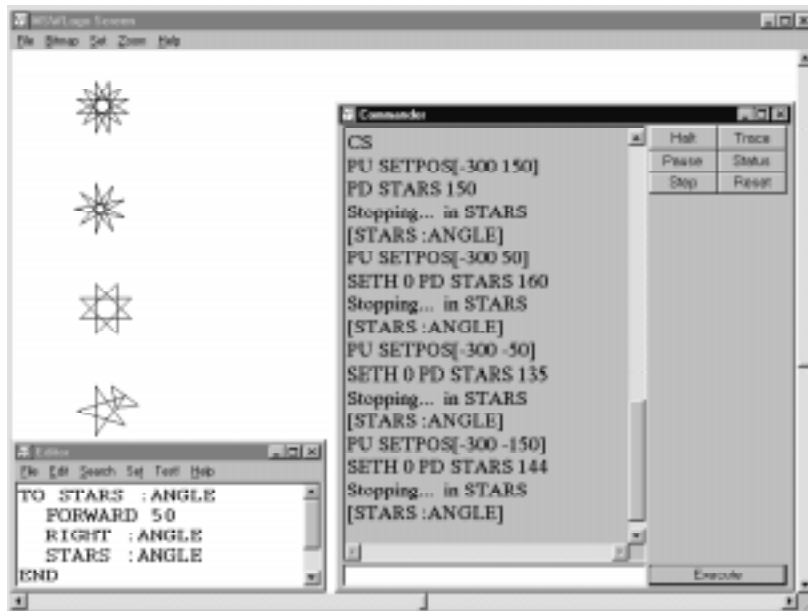
ในการเรียกตัวเองมาทำงานของกระบวนการ SPINSQUARES : ANGLE จะใส่ค่าตัวแปรด้วย ดังนั้นในการสร้างกระบวนการซึ่งมีตัวแปรรับเข้าและเป็นการทำซ้ำ ต้องแน่ใจว่า ใส่ชื่อตัวแปรในการเรียกใช้ตัวเองด้วย

กิจกรรมที่ 5.4

การใช้ตัวแปรเปลี่ยนแปลงรูปดาว

ต้องการสร้างกระบวนการสำหรับวาดรูปดาวที่มีตัวแปรเพื่อสามารถเปลี่ยนแปลงลักษณะตามต้องการ ดังนี้

- 1) สร้างกระบวนการ STARS :ANGLE ตามลำดับคำสั่งในหน้าต่าง Editor ดังรูปที่ 5.10



รูปที่ 5.10 กระบวนการ STARS :ANGLE และการใช้ตัวแปรสร้างดาวที่มีรูปร่างต่างกัน

- 2) สั่งเต่าให้ทำงานตามคำสั่งในกระบวนการ STARS :ANGLE โดยใส่ค่า ANGLE เป็น 150 160 135 และ 144 ตามลำดับคำสั่งในหน้าต่างบังคับการตามรูปที่ 5.10
- 3) เปรียบเทียบผลลัพธ์ของนักเรียนกับรูปที่ 5.10

หมายเหตุ คำสั่ง SETH ย่อมาจาก SETHEADING เป็นคำสั่งกำหนดตำแหน่งหัวเต่าให้หันชี้ไปยังมุมที่กำหนด มีรูปแบบคำสั่งคือ SETH angle เมื่อ angle คือมุมที่ใช้กำหนดให้หัวเต่าชี้

จากรูปที่ 5.10 คำสั่ง SETH 0 ใช้สั่งให้เต่าหันหัวชี้ไปทางทิศเหนือ แล้วจึงสั่งให้เต่าทำตามกระบวนการ STARS จะทำให้รูปที่ได้ทุกรูปเริ่มต้น โดยมีหัวเต่าชี้ทิศเหนือ

5.6 กระบวนการที่มีตัวแปรมากกว่า 1 ตัว

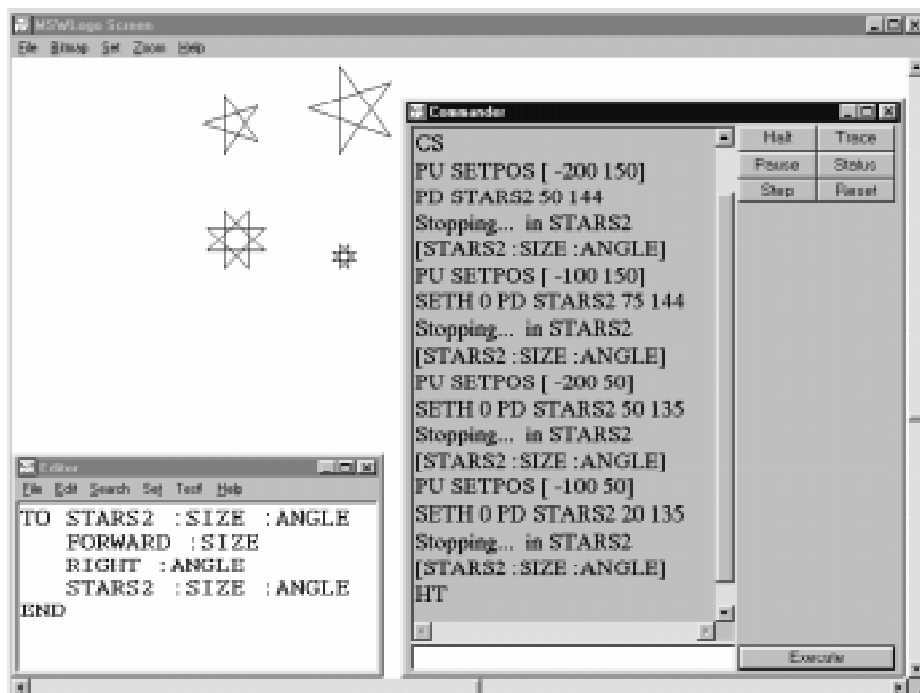
การกำหนดตัวแปรให้กับกระบวนการนั้นสามารถกำหนดตัวแปรได้มากกว่า 1 ตัว โดยการเติม :ชื่อตัวแปรต่อกันไปเรื่อย ๆ หลังชื่อกระบวนการ เช่น STARS2 :SIZE :ANGLE

กิจกรรมที่ 5.5

กระบวนการสร้างดาวที่มีตัวแปรมากกว่า 1 ตัว

ต้องการสร้างกระบวนการวาดรูปดาวที่มีตัวแปรสำหรับเปลี่ยนแปลงขนาดและมุม โดยกำหนดให้กระบวนการชื่อ STARS2 มีตัวแปร 2 ตัว สำหรับกำหนดขนาดและมุมเป็น SIZE และ ANGLE ตามลำดับ ให้ทำตามขั้นตอนดังนี้

- 1) สร้างกระบวนการ STARS2 :SIZE :ANGLE ตามลำดับคำสั่งในหน้าต่าง Editor ดังรูปที่ 5.11



รูปที่ 5.11 กระบวนการ STARS2 :SIZE :ANGLE และผลการทำงาน

- 2) สั่งเต่าให้ทำงานตามคำสั่งในกระบวนความ STARS2 :SIZE :ANGLE ตามลำดับคำสั่งดังนี้
- STARS2 50 144
 STARS2 75 144
 STARS2 50 135
 STARS2 20 135
- 3) เปรียบเทียบผลลัพธ์ของนักเรียนกับรูปที่ 5.11
- 4) สมมุติว่าส่วนแสดงผลเป็นห้องฟ้า ให้นักเรียนวาดดาวรูปร่างต่าง ๆ หลาย ๆ ขนาด และมีสีต่าง ๆ กัน จัดระดับบนห้องฟ้านี้ตามจินตนาการของนักเรียน

คำแนะนำ ให้ใช้คำสั่งต่อไปนี้ช่วยในการสร้างภาพ

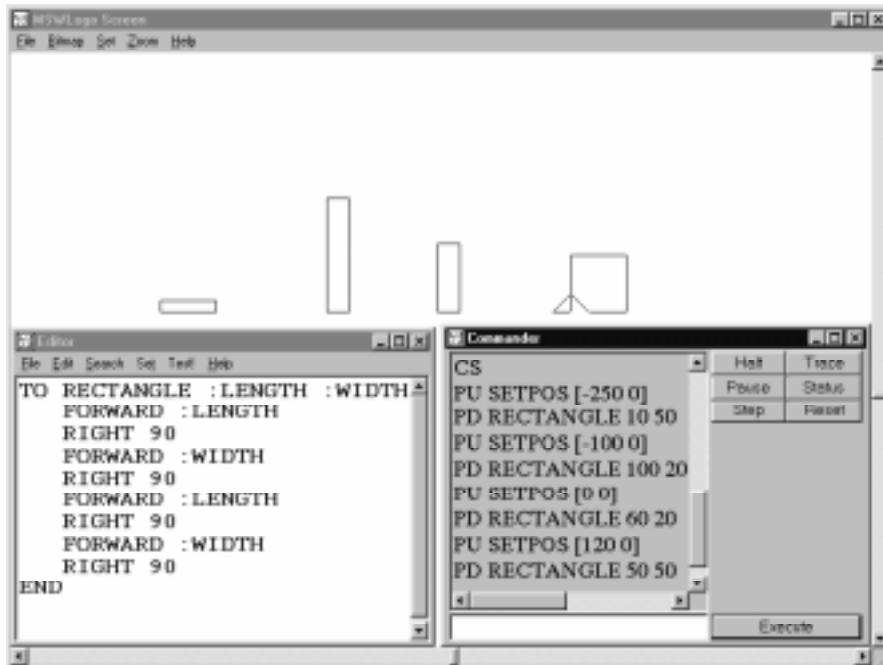
SETSC [a b c] กำหนดสีพื้น เช่น SETSC [0 0 0] กำหนดสีพื้นเป็นสีดำ
 SETPC [a b c] กำหนดสีเส้น
 SETPOS [x y] กำหนดตำแหน่งเต่า
 SETH n กำหนดทิศของหัวเต่า

กิจกรรมที่ 5.6

กระบวนความวาดสี่เหลี่ยมผืนผ้า

ต้องการสร้างกระบวนความวาดสี่เหลี่ยมผืนผ้าชื่อ RECTANGLE มีตัวแปร 2 ตัว สำหรับกำหนดความยาวและความกว้างเป็น ENGTH และ WIDTH ให้ทำตามขั้นตอนดังนี้

- 1) สร้างกระบวนความ RECTANGLE :LENGTH :WIDTH ซึ่งประกอบด้วยลำดับคำสั่งในหน้าต่าง Editor รูปที่ 5.12



รูปที่ 5.12 กระบวนการความ RECTANGLE :LENGTH :WIDTH และผลการทำงาน

- 2) สั่งเต่าให้ทำงานตามคำสั่งในกระบวนการความ RECTANGLE :LENGTH :WIDTH ดังนี้
 - RECTANGLE 10 50
 - RECTANGLE 100 20
 - RECTANGLE 60 20
 - RECTANGLE 50 50
- 3) เปรียบเทียบผลลัพธ์ของนักเรียนกับรูปที่ 5.12

5.7 กระบวนการความย่อที่มีตัวแปร

ถ้าพิจารณากระบวนการความต่าง ๆ ที่ได้เรียนรู้มาแล้ว จะพบว่าคำสั่งบางคำสั่งในกระบวนการเป็นกระบวนการ เรียกว่ากระบวนการความย่อ กระบวนการความย่อที่ต้องการกำหนดค่าให้สามารถเปลี่ยนแปลงได้ จะกำหนดให้มีตัวแปรใช้ในกระบวนการความย่อ เช่น

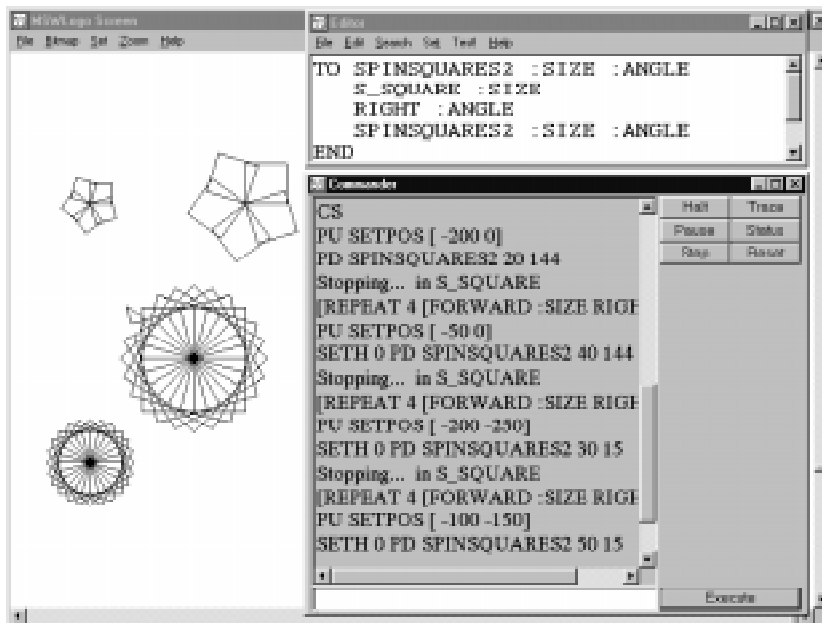
กระบวนการความ SPINSQUARES :ANGLE ในหัวข้อที่ 5.5 มีคำสั่งที่เป็นกระบวนการความย่อคือกระบวนการ S_SQUARE 50 ที่มีการกำหนดค่าคงที่ ถ้าต้องการให้สามารถเปลี่ยนแปลงค่า สามารถกำหนดให้กระบวนการความย่อมีตัวแปรรับค่าได้ ตามตัวอย่างในกิจกรรมต่อไป

กิจกรรมที่ 5.7

กระบวนการความย่อที่มีตัวแปร

ต้องการแก้ไขกระบวนการความย่อ SPINSQUARES :SIZE :ANGLE ให้มีการเรียกใช้กระบวนการความย่อ S_SQUARE :SIZE สำหรับกำหนดค่าตัวแปรต่าง ๆ โดยให้กระบวนการความย่อชื่อ SPINSQUARES2 :SIZE :ANGLE และกระบวนการความย่อชื่อ S_SQUARE :SIZE ทำได้ตามขั้นตอนดังนี้

- 1) สร้างกระบวนการความย่อ SPINSQUARES2 :SIZE :ANGLE ตามลำดับคำสั่งในรูปที่ 5.13



รูปที่ 5.13 กระบวนการความย่อ SPINSQUARES2 :SIZE :ANGLE และผลการทำงาน

- 2) สั่งเต้าให้ทำงานตามคำสั่งในกระบวนการความย่อ SPINSQUARES2 :SIZE :ANGLE โดยใส่ค่าตัวแปร SIZE และ ANGLE เป็นขนาดและมุมต่าง ๆ กัน ดังนี้

SPINSQUARES2 20 144

SPINSQUARES2 40 144

SPINSQUARES2 30 15

SPINSQUARES2 50 15

- 3) เปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.13

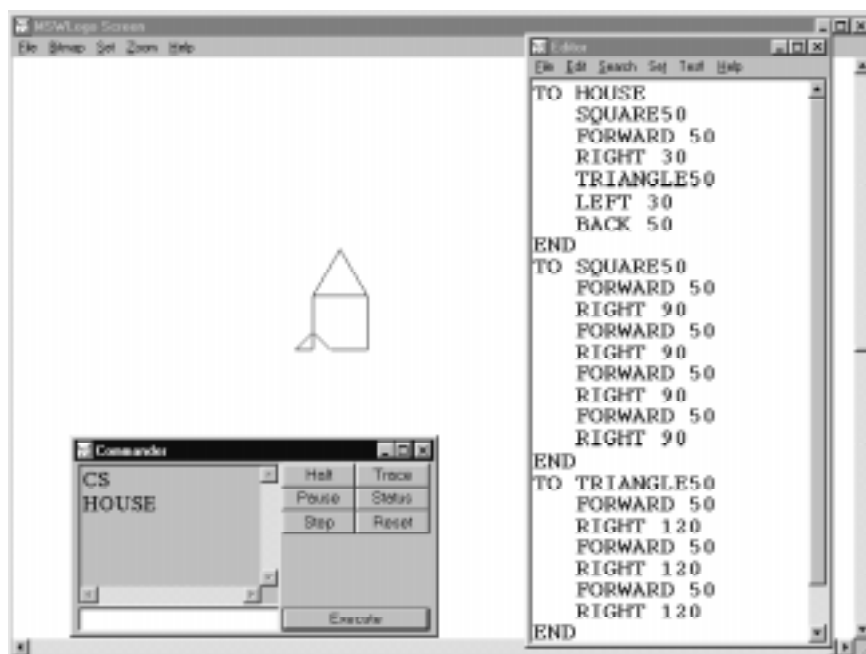
กิจกรรมที่ 5.8

การใช้กระบวนการความย่อवादรูปบ้าน

ต้องการวาดรูปบ้านโดยใช้กระบวนการความย่อवादรูปบ้านที่มีกระบวนการความย่อ อดตามขั้นตอนดังนี้

1) การวาดรูปบ้านที่มีขนาดคงที่

- 1.1) สร้างกระบวนการความ HOUSE สำหรับสั่งเต้าवादรูปบ้านที่มีขนาดคงที่ ซึ่งประกอบด้วยกระบวนการความย่อवादรูปสี่เหลี่ยม SQUARE50 และกระบวนการความย่อवादรูปสามเหลี่ยม TRIANGLE50 ที่มีความยาวของด้านคงที่เท่ากับ 50 หน่วย ตามลำดับคำสั่งในรูปที่ 5.14

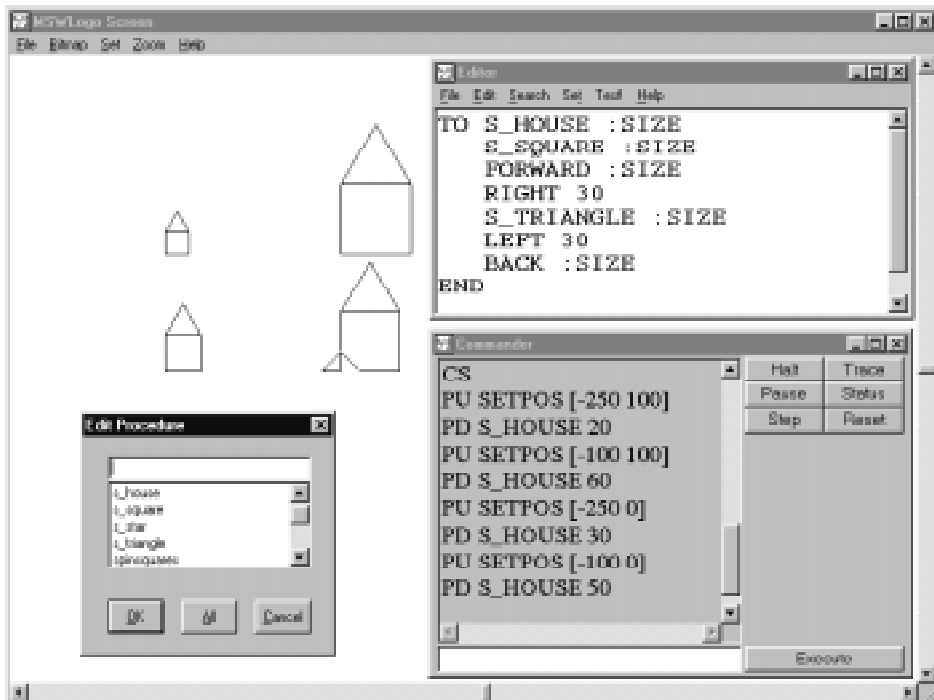


รูปที่ 5.14 กระบวนการความวาดรูปบ้านที่มีขนาดคงที่และผลที่ได้

- 1.2) สั่งเต้าवादรูปบ้านด้วยคำสั่ง HOUSE เปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.14

2) การวาดรูปบ้านที่มีขนาดตามต้องการ

- 2.1) สร้างกระบวนการ S_HOUSE :SIZE สำหรับสั่งเต๋าวาดรูปบ้านที่มีขนาดตามต้องการ โดยใช้กระบวนการย่อยที่สร้างไว้ในกิจกรรมก่อนหน้านี้ คือ กระบวนการวาดรูปสี่เหลี่ยม S_SQUARE :SIZE และกระบวนการย่อยวาดรูปสามเหลี่ยม S_TRIANGLE :SIZE ที่มีตัวแปร SIZE ใช้ร่วมกันเป็นตัวกำหนดความยาวของด้าน ตามลำดับคำสั่งในรูปที่ 5.15



รูปที่ 5.15 กระบวนการวาดรูปบ้านที่มีขนาดตามที่กำหนดและผลที่ได้

- 2.2) สั่งให้เต๋าทำงานตามคำสั่งในกระบวนการ S_HOUSE โดยใช้ค่าตัวแปร SIZE เป็นค่าต่างกัน ดังนี้

S_HOUSE 20

S_HOUSE 60

S_HOUSE 30

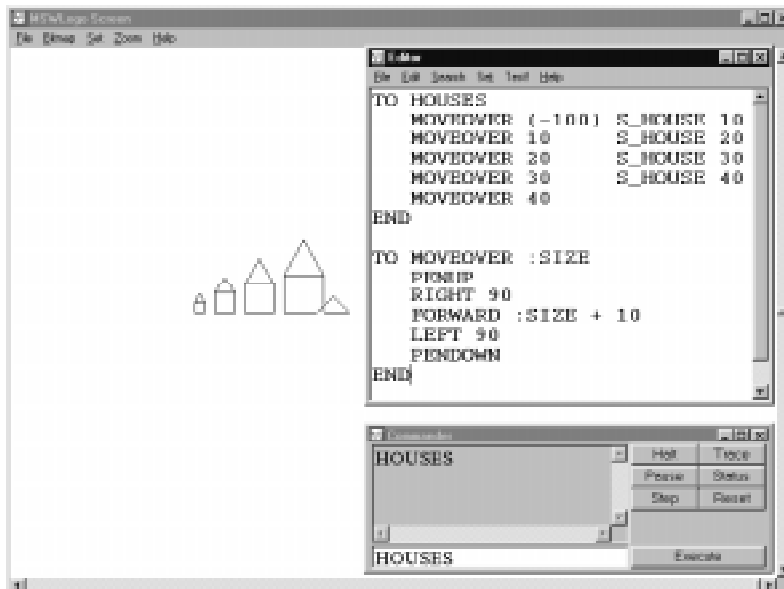
S_HOUSE 50

แล้วเปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.15

หมายเหตุ โปรดสังเกตรูปที่ 5.15 ในหน้าต่าง Edit Procedure จะปรากฏรายชื่อกระบวนการ ความ S_SQUARE และ กระบวนการความ S_TRIANGLE ซึ่งแสดงว่า เตารู้จักกระบวนการ ทั้งสองเพราะถูกจำไว้ในหน่วยความจำขณะทำงานแล้ว จึงสามารถใช้เป็นคำสั่งในกระบวนการ ความ S_HOUSE ได้

3) การวาดรูปบ้านที่มีขนาดตามต้องการ ณ ตำแหน่งที่กำหนด

3.1) สร้างกระบวนการ HOUSES สำหรับสั่งเตาวาดรูปบ้านที่มีขนาดตามต้องการ ณ ตำแหน่งที่กำหนด ซึ่งประกอบด้วยกระบวนการย่อยวาดรูปบ้านตามขนาดที่ต้องการ S_HOUSE :SIZE จากขั้นตอนก่อนและกระบวนการย่อยสำหรับย้ายตำแหน่งเตา MOVEOVER :UNIT ตามลำดับคำสั่งในรูปที่ 5.16



รูปที่ 5.16 บ้านเป็นแถวที่สร้างโดยกระบวนการ HOUSES

จากรูป กระบวนการ HOUSES เรียกใช้กระบวนการย่อยชื่อว่า MOVEOVER โดยในบรรทัดแรกของกระบวนการ HOUSES ค่าตัวแปรของ MOVEOVER เป็นค่าลบ เพื่อกำหนดค่าให้เตาเลื่อนตำแหน่งไปทางซ้ายแทนที่จะเลื่อนไปทางขวา ถ้าค่าของตัวแปรเป็นลบต้องใส่วงเล็บ เพราะถ้าไม่มีวงเล็บ เตาอาจคิดว่า สั่งให้นำค่าที่กำหนดให้ไปลบออกจากค่าใดค่าหนึ่ง

ให้สังเกตในกระบวนการ MOVEOVER บรรทัดคำสั่ง FORWARD :SIZE + 10 จะเห็นว่าสามารถใช้เครื่องหมายบวกเพื่อเพิ่มค่าของตัวแปร SIZE ได้ และในการทำงานเดียวกันสามารถใช้เครื่องหมายทางคณิตศาสตร์อื่น ๆ เช่น ลบ คูณ หาร ในการลดหรือเพิ่มค่าตัวแปรได้

- 3.2) สั่งเต้าให้ทำงานตามคำสั่งในกระบวนการ HOUSES แล้วเปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.16

กิจกรรมที่ 5.9

กระบวนการวาดรูปต้นไม้

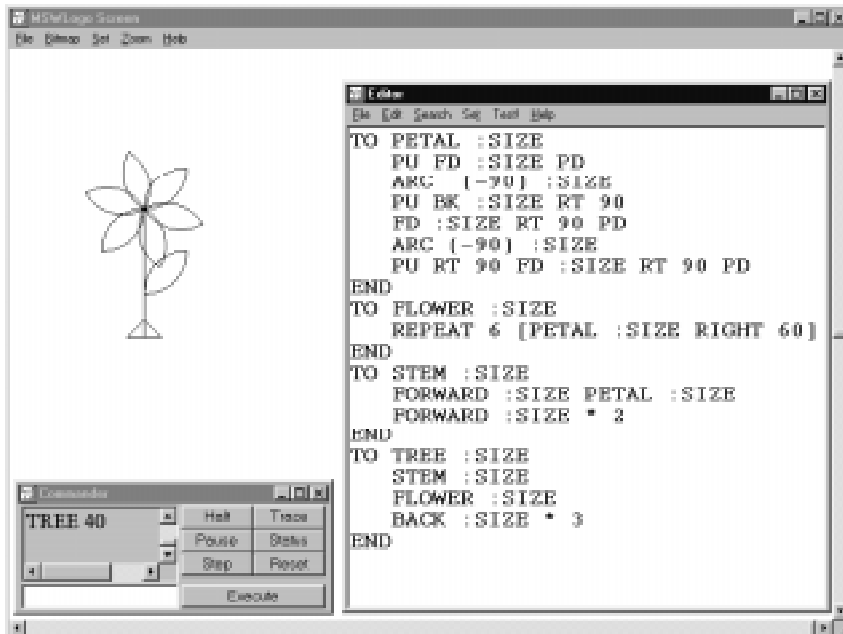
- 1) สร้างกระบวนการวาดรูปต้นไม้ชื่อ TREE ซึ่งประกอบด้วยกระบวนการย่อยต่าง ๆ ดังนี้

กระบวนการ PETAL ใช้สำหรับวาดภาพกลีบดอกไม้หรือกลีบใบ

กระบวนการ FLOWER ใช้สำหรับวาดภาพดอกไม้ 1 ดอก

กระบวนการ STEM ใช้สำหรับวาดภาพลำต้นและใบ

ตามลำดับคำสั่งในรูปที่ 5.17



รูปที่ 5.17 กระบวนการวาดรูปต้นไม้

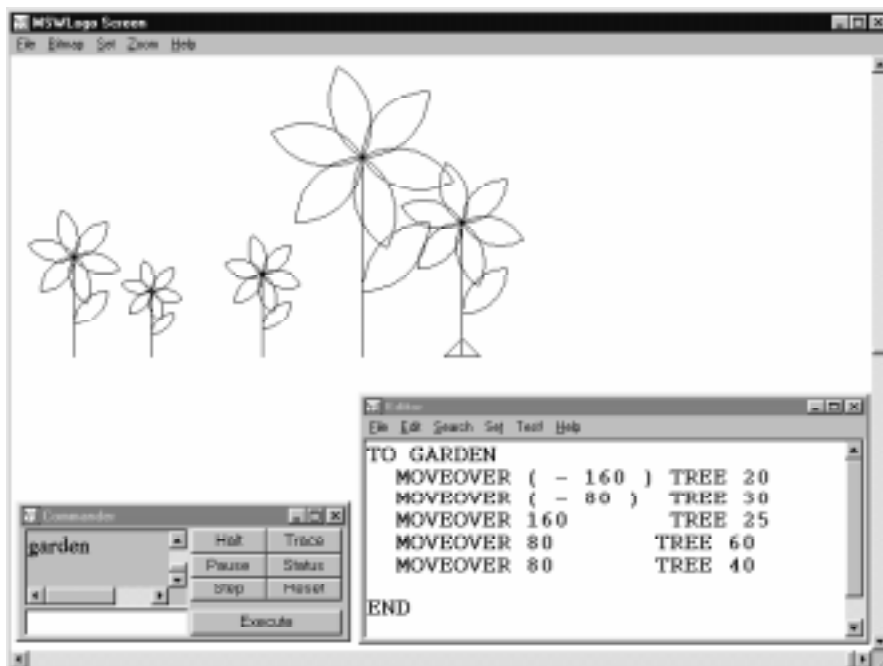
จากรูปในกระบวนการความย่อ STEM และ TREE มีการใช้เครื่องหมาย * แทนเครื่องหมายของการคูณ ดังนั้นคำสั่ง :SIZE * 2 หมายถึงให้คุณค่าตัวแปร SIZE ด้วย 2

- 2) สั่งให้เต่าวาดรูปโดยใช้คำสั่งกระบวนการ TREE 40 แล้วเปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.17

กิจกรรมที่ 5.10

กระบวนการสร้างสวนดอกไม้

สร้างกระบวนการ GRADEN สำหรับสั่งให้เต่าวาดสวนดอกไม้ดังกล่าวหรือในลักษณะอื่นตามความคิดของตนเอง โดยใช้กระบวนการ TREE เป็นกระบวนการย่อย ดังตัวอย่างตามลำดับคำสั่งในรูปที่ 5.18

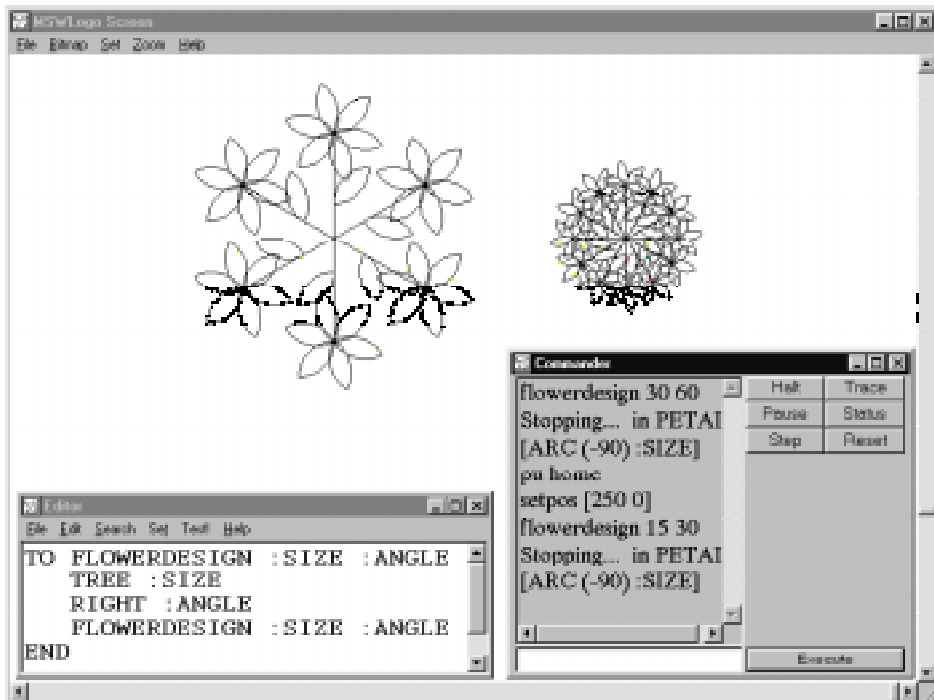


รูปที่ 5.18 สวนดอกไม้ที่มีดอกไม้ขนาดต่าง ๆ

กิจกรรมที่ 5.11

กระบวนการความวาดชื่อดอกไม้

- 1) สร้างกระบวนการ FLOWERDESIGN :SIZE :ANGLE สำหรับสั่งให้เต่าวาดภาพชื่อดอกไม้ โดยใช้กระบวนการ TREE เป็นกระบวนการย่อย ดังตัวอย่างตามลำดับคำสั่งในรูปที่ 5.19



รูปที่ 5.19 กระบวนการ FLOWERDESIGN :SIZE :ANGLE และผลที่ได้

- 2) สั่งเต่าให้ทำงานตามคำสั่งในกระบวนการ FLOWERDESIGN :SIZE :ANGLE โดยใส่ค่าตัวแปร SIZE และ ANGLE เป็นค่าต่างกัน ดังนี้

FLOWERDESIGN 30 60

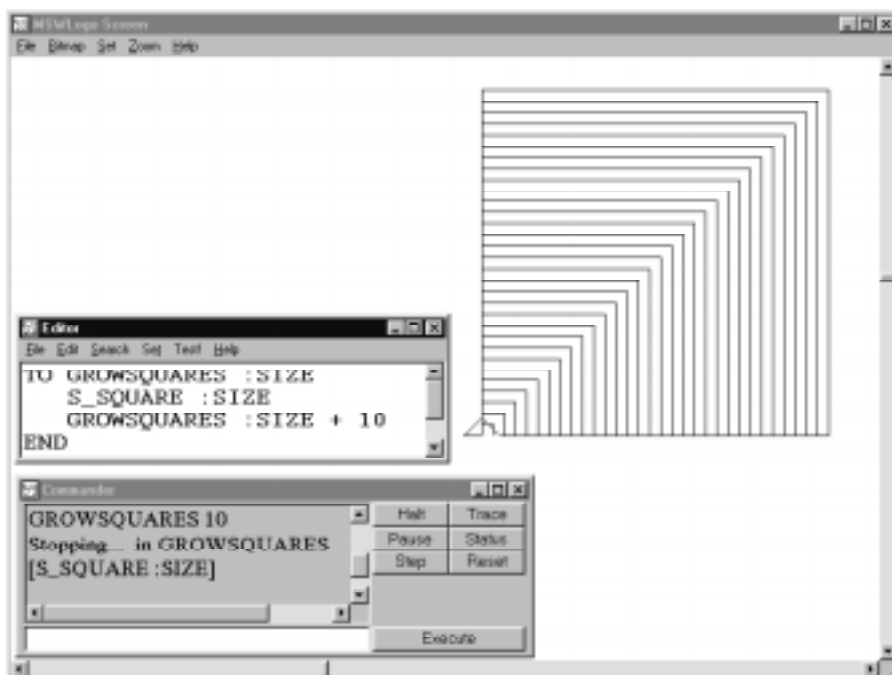
FLOWERDESIGN 15 30

แล้วเปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.19

กิจกรรมที่ 5.12

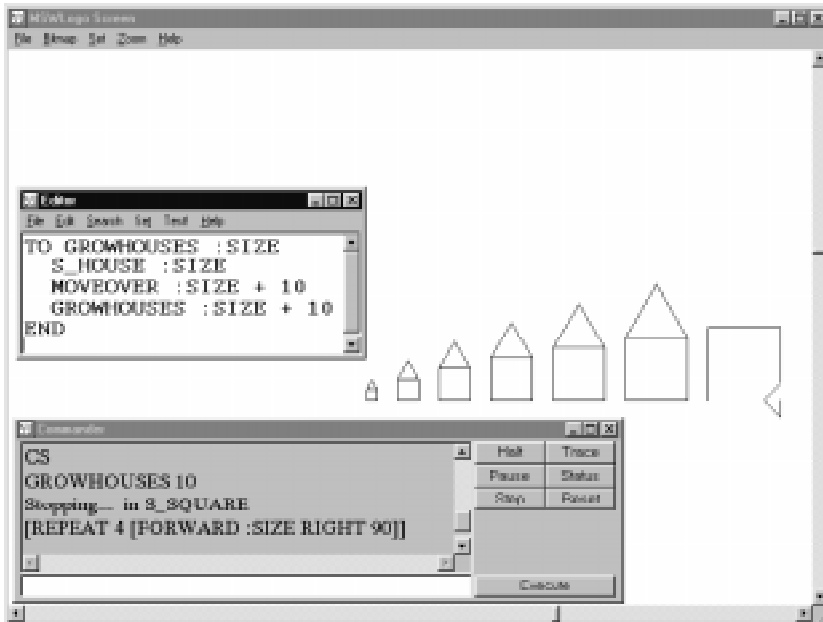
กระบวนการวาดภาพที่มีขนาดโตขึ้นเรื่อยๆ

- 1) สร้างกระบวนการ GROWSQUARES :SIZE สำหรับสั่งให้เต่าวาดภาพสี่เหลี่ยมที่มีขนาดโตขึ้นเรื่อย ๆ โดยใช้กระบวนการ S_SQUARE เป็นกระบวนการย่อย ตามลำดับคำสั่งในรูปที่ 5.20 แล้วสั่งเต่าวาดภาพด้วยคำสั่ง GROWSQUARES 10 และให้หยุดการวาดภาพของเต่าโดยคลิกปุ่ม Halt ดูผลลัพธ์ที่ได้



รูปที่ 5.20 กระบวนการ GROWSQUARES :SIZE และผลที่ได้

- 2) สร้างกระบวนการ GROWHOUSES :SIZE สำหรับสั่งให้เต่าวาดภาพบ้านที่มีขนาดโตขึ้นเรื่อย ๆ ตามลำดับคำสั่งในรูปที่ 5.21 แล้วสั่งเต่าวาดภาพด้วยคำสั่ง GROWHOUSES 10 และให้หยุดการวาดภาพของเต่าโดยคลิกปุ่ม Halt ดูผลลัพธ์ที่ได้



รูปที่ 5.21 กระบวนการความ GROWHOUSES :SIZE และผลที่ได้

- 3) ให้พิจารณาคำสั่งและผลลัพธ์จากกระบวนการทั้งสองแล้วตอบคำถามต่อไปนี้
 - 3.1) ผลลัพธ์ของการใช้กระบวนการ CROWSQUARES คือ
 - 3.2) ผลลัพธ์ของการใช้กระบวนการ CROWSQUARES คือ
 - 3.3) คำสั่งที่ให้เตาวาดรูปทั้งสองคือ
 - 3.4) ขนาดของสี่เหลี่ยมที่เล็กที่สุดคือ ด้วยคำสั่ง.....
 ขนาดของบ้านที่เล็กที่สุดคือ ด้วยคำสั่ง.....
 - 3.5) รูปสี่เหลี่ยมมีขนาดเพิ่มขึ้นรูปละ ด้วยคำสั่ง.....
 รูปบ้านมีขนาดเพิ่มขึ้นรูปละ ด้วยคำสั่ง.....
 - 3.6) กระบวนการย่อยที่ถูกเรียกใช้ในกระบวนการทั้งสอง
 กระบวนการGROWSQUARE มี.....
 กระบวนการ GROWHOUSE มี
 - 3.7) คำสั่งที่หยุดการทำงานของกระบวนการทั้งสอง คือ

5.8 การหยุดการทำงานของกระบวนความ

การหยุดการทำงานของเต่าในกิจกรรมที่ 5.12 ทำโดยคลิกปุ่ม Halt ซึ่งหมายถึงการสั่งหยุดการทำงานของโปรแกรมในขณะที่ทำงาน ไม่ใช่การหยุดทำงานเมื่อสิ้นสุดคำสั่งตามปกติ ซึ่งไม่เหมาะกับการนำไปใช้งานในกระบวนความทั่วไป ในกิจกรรมที่ 5.13 จะเป็นการควบคุมให้กระบวนความสิ้นสุดการทำงานด้วยคำสั่งในกระบวนความ โดยกำหนดเงื่อนไขการหยุดทำงาน คำสั่งที่ใช้คือคำสั่ง IF ซึ่งมีรูปแบบดังนี้

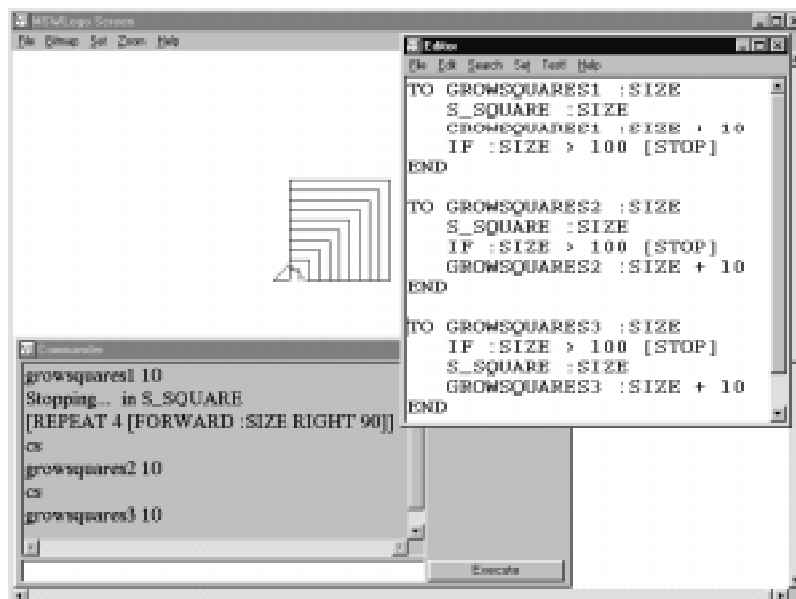
IF เงื่อนไข [STOP]

ถ้าต้องการให้กระบวนความหยุดทำงานเมื่อค่าตัวแปร SIZE มากกว่า 100 หน่วย คำสั่งที่ต้องใช้ คือ IF :SIZE > 100 STOP

กิจกรรมที่ 5.13

การหยุดการวาดภาพที่มีขนาดโตขึ้นเรื่อยๆ

- 1) สร้างกระบวนความ GROWSQUARES1 :SIZE กระบวนความ GROWSQUARES2 :SIZE และกระบวนความ GROWSQUARES3 :SIZE ตามลำดับคำสั่งในรูปที่ 5.22



รูปที่ 5.22 คำสั่ง IF ในกระบวนความ GROWSQUARES :SIZE และผลที่ได้

- 2) แล้วสั่งเต๋าวาดภาพด้วยคำสั่ง GROWSQUARES1 10
GROWSQUARES2 10
และ GROWSQUARES3 10
- 3) เปรียบเทียบผลลัพธ์ที่ได้จะพบว่า
ผลลัพธ์ที่ได้จากการใช้คำสั่ง GROWSQUARES1 10 คือ
ผลลัพธ์ที่ได้จากการใช้คำสั่ง GROWSQUARES2 10 คือ
ผลลัพธ์ที่ได้จากการใช้คำสั่ง GROWSQUARES3 10 คือ
- 4) คำสั่งที่แตกต่างในกระบวนความทั้งสามคือ
.....

หมายเหตุ

(1) จากผลลัพธ์ที่ได้กิจกรรมที่ 5.13 แสดงว่าตำแหน่งการใส่เงื่อนไขเป็นสิ่งสำคัญในการควบคุมการทำงานของกระบวนความ

(2) การใส่เงื่อนไขใน GROWSQUARES1 ไม่สามารถใช้ได้เพราะเต๋าย่อมไม่มีโอกาสไปทำคำสั่งเงื่อนไขหยุดการทำงานในบรรทัดที่ 4 ของกระบวนความเลย เพราะคำสั่งในบรรทัดที่ 3 จะเรียก GROWSQUARES1 มาใช้ใหม่ คือ เริ่มต้นใหม่เรื่อย ๆ

(3) ใน GROWSQUARES2 จะหยุดเมื่อมีการวาดรูปที่ใหญ่กว่า 100 แล้ว เพราะว่าจะวาดรูปก่อน แล้วจึงตรวจดูว่ามีขนาดใหญ่กว่า 100 หรือไม่

(4) ใน GROWSQUARES3 จะตรวจดูว่ารูปขนาดใหญ่กว่า 100 หรือไม่ก่อนแล้วจึงวาดรูป เพราะฉะนั้นจะหยุดก่อน

(5) หลักสำคัญที่สุดของโปรแกรมข้อหนึ่งคือการตรวจดูว่าจะหยุดทำงานเมื่อไร โดยตรวจเงื่อนไขว่าเป็นจริงหรือเท็จ ถ้าเป็นเท็จจะเปลี่ยนค่าตัวแปรและทำงานต่อไป ถ้าเป็นจริงก็จะหยุดการทำงาน วิธีที่ดีที่สุดในการตรวจสอบ คือ การไล่ดูคำสั่งทีละบรรทัด โดยสมมติว่าเราคือเต๋ายิ่งทำหน้าที่ปฏิบัติตามคำสั่งทีละบรรทัด แล้วจะได้ผลลัพธ์อย่างไร

ในกิจกรรมที่ 5.7 กระบวนการความ SPINSQUARES2 ใช้ตัวแปร 2 ตัวแปร ดังนี้

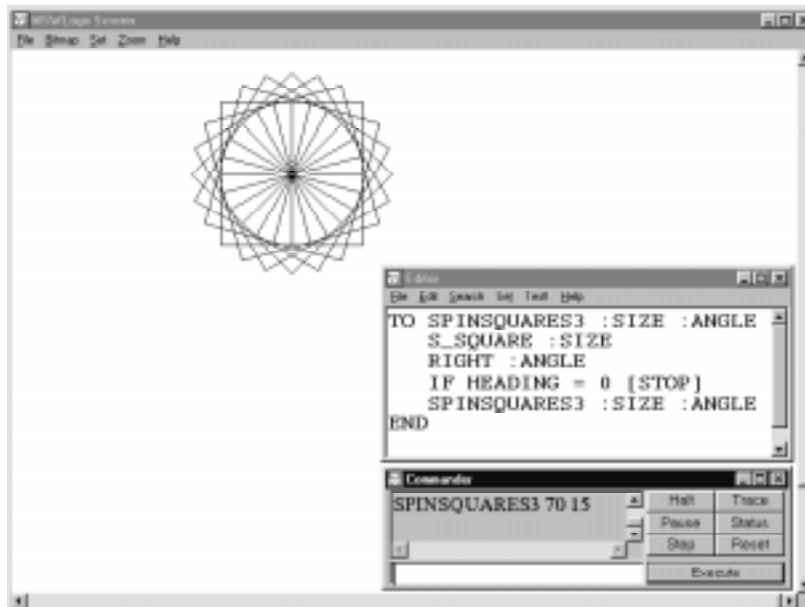
```
TO SPINSQUARES2 :SIZE :ANGLE
  S_SQUARE :SIZE
  RIGHT :ANGLE
  SPINSQUARES2 :SIZE :ANGLE
END
```

กระบวนการความนี้จะเรียกตัวเองมาทำงานไม่สิ้นสุด หากต้องการกำหนดคำสั่งเงื่อนไขให้เต่าหยุดวาดรูปหลังจากวาดเสร็จ 1 รูป โดยใช้จำนวนของรูปสี่เหลี่ยมย่อมกระทำได้ยากเพราะจำนวนของรูปสี่เหลี่ยมที่ใช้ในการวาดรูป 1 รูป ขึ้นกับมุมที่กำหนดให้ เป็นค่าที่ไม่แน่นอน จึงต้องคิดหาเงื่อนไขอื่น และเมื่อพิจารณารูปจะพบว่า เต่าวาดรูปเสร็จ 1 รูป เมื่อเต่าหันหัวชี้ไปยังมุมเดียวกับมุมเริ่มต้น ดังนั้นเงื่อนไขการหยุดที่ง่ายที่สุดคือ ให้หยุดเมื่อหัวเต่าชี้กลับไปจุดเริ่มต้น ในกรณีนี้จะหยุดได้โดยให้ตัวเต่าคือ HEADING เป็น 0 อีกครั้งหนึ่ง

กิจกรรมที่ 5.14

การหยุดการวาดภาพ

1) สร้างกระบวนการความ SPINSQUARES3 :SIZE :ANGLE ตามลำดับคำสั่งในรูปที่ 5.23



รูปที่ 5.23 คำสั่งในกระบวนการความ SPINSQUARES3 และผลที่ได้

- 2) สั่งให้เต่าวาดรูปโดยใช้คำสั่ง SPINSQUARES3 70 15 เปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.23

หมายเหตุ กระบวนการ SPINSQUARES3 จะหยุดหลังจากวาดรูปเสร็จ 1 รูป ซึ่งเป็นผลมาจากการทำงานของคำสั่ง IF HEADING = 0 [STOP]

HEADING เป็นคำสั่งในภาษาโลโก้ที่ให้ค่ามุมซึ่งเป็นทิศทางที่หัวเต่าชี้ในขณะนั้น ค่ามุมนี้เริ่มต้นจาก 0 ถึง 360 องศา หมุนตามเข็มนาฬิกา ดูตัวอย่างต่อไป

พิมพ์กระบวนการต่อไปนี แล้วดูผลลัพธ์

```
CS
RIGHT 45
PRINT HEADING
45
RIGHT 90
PRINT HEADING
135
```

เครื่องหมาย = เป็นคำสั่งในภาษาโลโก้ด้วย และต้องการค่า 2 ค่ามาเปรียบเทียบว่าเท่ากันหรือไม่ ซึ่งจะส่งคำตอบว่า TRUE ในกรณีที่ทั้ง 2 ค่า เท่ากันและจะส่งคำตอบ FALSE ในกรณีที่ 2 ค่าต่างกัน เช่น

```
PRINT 5 = 4 + 1
TRUE
PRINT 3 = 0
FALSE
```

IF เป็นคำสั่งในภาษาโลโก้ที่ต้องการค่าใดค่าหนึ่งคือ TRUE หรือ FALSE เป็นข้อมูลรับเข้า ถ้า IF รับ TRUE เป็นข้อมูลรับเข้า คอมพิวเตอร์จะทำคำสั่งของบรรทัดนั้นต่อไป ถ้ารับข้อมูลรับเข้าเป็น FALSE จะไม่ทำคำสั่งในบรรทัดนั้น แต่จะเลื่อนทำบรรทัดต่อไป

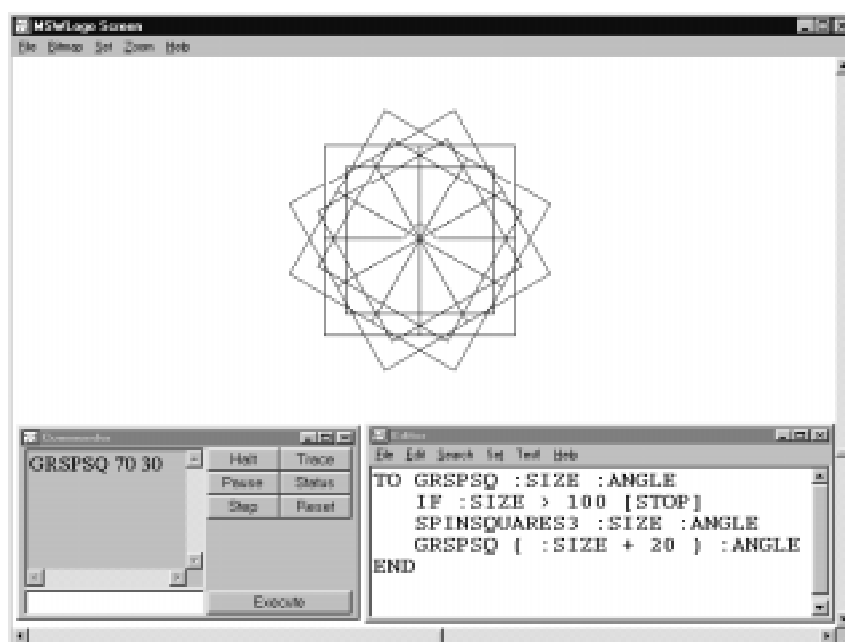
IF HEADING = 0 [STOP] หมายถึง ถ้าค่าจาก HEADING มีค่าเท่ากับ 0 จะได้ค่า TRUE ที่ IF ทำให้คำสั่ง STOP ทำงาน แต่ถ้าค่าจาก HEADING ไม่เท่ากับ 0 จะได้ค่า FALSE ที่ IF ให้ข้ามไปทำคำสั่งในบรรทัดถัดไป

เมื่อกระบวนการที่ถูกเรียกจบลงแล้วจะทำคำสั่งต่อไปจากบรรทัดข้างบน ในแต่ละกระบวนการจะคอยจนกว่ากระบวนการที่ถูกเรียกจะหยุด แล้วจึงทำบรรทัดต่อไป

กิจกรรมที่ 5.15

การหยุดการวาดภาพในกระบวนการ GRSPSQ

- 1) สร้างกระบวนการ GRSPSQ (GROW-SPIN-SQUARES) ตามลำดับคำสั่งในรูปที่ 5.24



รูปที่ 5.24 คำสั่งในกระบวนการ GRSPSQ :SIZE :ANGLE และผลที่ได้

- 2) วาดรูปโดยใช้คำสั่ง GRSPSQ 70 30 เปรียบเทียบผลลัพธ์ที่ได้กับรูปที่ 5.24
- 3) ใช้คำสั่ง GRSPSQ โดยเปลี่ยนค่ามุมเป็นค่าต่าง ๆ เปรียบเทียบภาพที่ได้กับเพื่อน ๆ
- 4) ให้ตัดคำสั่ง STOP ออกจากกระบวนการนี้ แล้วใช้สั่งให้เต๋าวาดภาพ

ผลที่ได้คือ

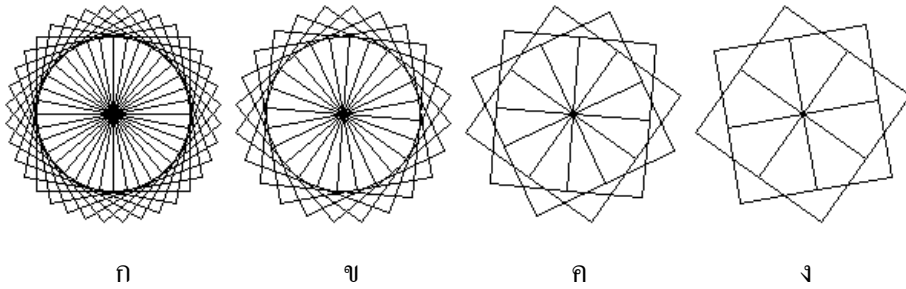
หมายเหตุ

(1) ในบรรทัดก่อนสุดท้ายจะเห็นวงเล็บที่เขียนเป็น (:SIZE + 20) ซึ่งไม่จำเป็นต้องใส่วงเล็บ แต่ที่ใส่วงเล็บเพราะว่าต้องการให้เข้าใจว่า (:SIZE + 20) เป็นข้อมูลรับเข้าอันหนึ่ง

(2) ค่า :ANGLE ที่ใส่จะเป็นตัวกำหนดรูปร่างของรูป SIZE จะได้รับการตัดสินใจโดยขนาดเริ่มต้น

แบบฝึกหัดบทที่ 5

1. จงใช้กระบวนการความ STARS2 :SIZE :ANGLE วาดดาว 9 แฉก 12 แฉก และ 18 แฉก หาวามุมที่ใช้มีค่าเท่าไร
2. จงใช้กระบวนการความ SPINSQUARES2 :SIZE :ANGLE วาดภาพต่าง ๆ ดังนี้



3. จงแก้ไขกระบวนการความ TREE :SIZE ให้เป็นกระบวนการความ TREE :SIZE :STYLE :COLOUR1 :COLOUR2 :COLOUR3 เพื่อวาดรูปดอกไม้ที่มีลักษณะต่าง ๆ กันตามค่าตัวแปร STYLE และมีสีต่าง ๆ กันตามค่าตัวแปร COLOUR1 COLOUR2 และ COLOUR3
4. จงแก้ไขกระบวนการความ FLOWERDESIGN :SIZE :ANGLE เพื่อสั่งให้โปรแกรมหยุดการทำงานด้วยคำสั่งในกระบวนการความ เมื่อวาดดอกไม้เสร็จ 1 ช่อ
5. จงสร้างกระบวนการความที่คล้าย GRSPSQ โดยใช้รูปสามเหลี่ยมแทนรูปสี่เหลี่ยมเดิม
6. จงเขียนกระบวนการความวาดรูปสวนดอกไม้ บ้าน และดวงดาว ให้มีสีสันต่าง ๆ โดยใช้กระบวนการความที่สร้างขึ้นในบทนี้ รวมทั้งอาจสร้างขึ้นใหม่ตามที่จำเป็น
7. จงใช้ความรู้ที่เรียนมาสร้างรูปตามความคิดของตนเองให้สวยงาม