

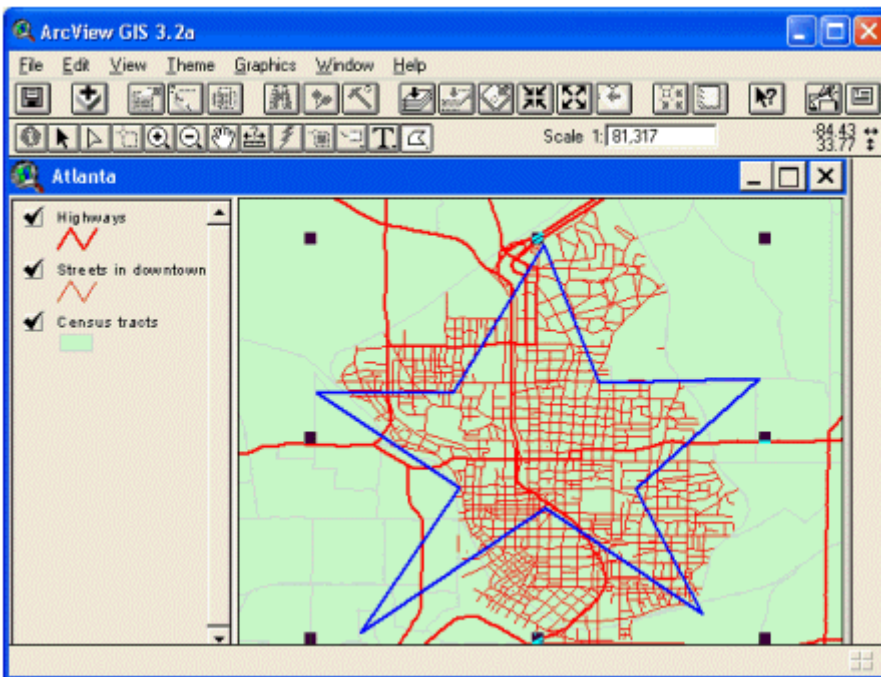
การสร้าง Graphic บน View window

- การสร้าง Graphic บน View window

**ดัดแปลงและเรียบเรียงจาก หนังสือ Introduction to AVENUE โดย ESRI**

โดย... อาจารย์สุเพชร จิระจรรกุล

บทความชุดนี้จะอธิบายถึงการทำงานด้วย Graphic ที่เราสามารถสร้างขึ้นบน ArcView ซึ่งเราจะต้องเข้าใจว่ารูปภาพเชิงเรขาคณิต (สี่เหลี่ยม, วงกลม วงรี เป็นต้น) ที่ประกอบด้วยค่าพิกัด x,y ที่จะสร้างขึ้นเป็นรูปทรงเรขาคณิตต่างๆ นั้นเราอาจจะเรียกได้ว่า Shape ซึ่งอาจจะแสดงอยู่ในรูปแบบจุด เส้น สี่เหลี่ยม วงกลม วงรี หรือหลายเหลี่ยม ก็ได้ทั้งหมด แต่ถ้า Graphic นั้นจะแสดงในรูปแบบสัญลักษณ์ที่คล้ายกับ Shape คืออาจจะอยู่ในรูปแบบสัญลักษณ์จุด สัญลักษณ์เส้น สัญลักษณ์สี่เหลี่ยม เป็นต้น โดยที่ทั้ง Shape และ Graphic นั้นสามารถแสดงผลบน View และ Layout ได้ทั้งสองชนิด ในขณะที่ Shape แสดงผลแบบชั่วคราว (เปิด/ปิด ได้) เท่านั้น แต่ Graphic จะแสดงผลแบบถาวรบน View หรือ Layout (ถ้าจะปิดต้องสั่งลบออก)



เราอาจจะสร้าง Shape และ Graphic ขึ้นมาได้ โดยเราวาดพื้นที่ที่เราสนใจใน View อาจจะเลือก Theme features ใดๆ อาจจะเพิ่ม Features ใหม่ และการเปลี่ยนสัญลักษณ์ของ Graphic ได้ เราอาจสร้าง Shape โดยผู้ใช้เป็นผู้นำเข้า โดยมีการโต้ตอบกับผู้ใช้โปรแกรมด้วยการให้ผู้ใช้นำเข้าพิกัดภูมิศาสตร์ที่ทราบค่า ผ่าน message box โดย Shape ที่สร้างได้อาจจะนำไปใช้ในการเลือก

Features หรืออาจใช้ในการสร้าง Graphic หรือการสร้าง Feature ต่างๆใน Theme หรือ แสดงผลในรูปแบบต่างๆ ได้

### การสร้าง Shape

ในการสร้าง Shape เราสามารถสร้างได้หลายวิธี หลายรูปแบบของข้อมูล ตัวอย่างเช่น ถ้าเรา ต้องการสร้างข้อมูลแบบ point shape เราจะต้องให้ค่าพิกัด 2 ตัว คือ x และ y เช่น

```
pointTHA = Point.Make(9428775, 1651000)
```

```
pointIND = 7220500@2457800
```

ในการสร้าง rectangle shape นั้นจะต้องมีจุดเริ่มต้น และจุดมุมตรงข้าม ซึ่งตัวอย่างเช่น

```
theRectTHA = Rect.Make(9458700@5000, 1745900@5000)
```

```
theRectTOT = Rect.Make(pointIND, pointTHA)
```

ถ้าต้องการสร้าง circle shape นั้นจะต้องให้ค่าจุดศูนย์กลาง และรัศมี ดังนี้

```
theCircleTHA = Circle.Make(9428775@1651000, 1000)
```

```
theCircleIND = Circle.Make(pointIND, 1000)
```

### การเปลี่ยนให้ Shape เป็น Graphic

เมื่อเราได้สร้าง Shapes บน View แล้วเราจะต้องเปลี่ยนให้ Shapes เหล่านั้นได้กลายเป็น Graphic เพื่อให้มันแสดงผลบน View หรือ layout มีขั้นตอนดังนี้

การทำให้ Shape เป็น Graphic โดยใช้รูปแบบคำสั่ง

```
thepointTHA = GraphicShape.Make(pointTHA)
```

จากนั้นให้รับค่ารายการ List ของ Graphics บน View

```
theView = av.GetActiveDoc
```

```
theGraphicList = theView.GetGraphics
```

จากนั้นจึงให้เพิ่ม Graphic ลงใน View ดังนี้

```
theGraphicList.Add(thePointTHA)
```

### การเปลี่ยนแปลงสัญลักษณ์ของ Graphic

รูปแบบ Graphic ทุกตัวอาจจะเปลี่ยนแปลงสัญลักษณ์ต่างๆ ได้ เช่น fill pattern, outline color, ขนาดของเส้น outline และคุณสมบัติอื่นๆ

อันดับแรกเราจะต้องให้รับค่าสัญลักษณ์ของ graphic ดังนี้

```
theSym = thePointTHA.GetSymbol หรือ
```

```
theSym = theCircleGraphic.GetSymbol
```

จากนั้นเราก็เพิ่มสีให้กับ graphic โดยมีการส่งค่าสีให้โดย ArcView มีสีมาตรฐานให้เลือกใช้งาน เช่น black, white, red, blue, cyan, yellow, green, magenta, gray เราจะต้องให้ค่าในชุดคำสั่งดังนี้

```
colorRed = Color.GetRed
```

```
colorGreen = Color.GetGreen
```

```
colorBlue = Color.GetBlue
```

ถ้าไม่พอใจสีข้างต้นอาจจะสร้างขึ้นมาเองได้โดยใช้คำสั่ง Color.Make ดังนี้

```
lightGreen = Color.Make
```

```
theColorList = {100,210,170}
```

```
lightGreen.SetRGBList(theColorList)
```

จากนั้นให้ทำการเปลี่ยนคุณสมบัติของ fill pattern ซึ่งรูปแบบในการ Set Style ของ RasterFill นั้นมีได้หลายรูปแบบคือ

#RASTERFILL\_STYLE\_EMPTY คือไม่ต้องใส่สีใดๆ

#RASTERFILL\_STYLE\_OPAQUESTIPPLE ต้องการแสดงสีเส้นพาดผ่านโดยมีสีของ background ด้วย

#RASTERFILL\_STYLE\_SOLID เป็นการให้สีทึบ

#RASTERFILL\_STYLE\_STIPPLE เป็นการให้สีเส้นพาดผ่าน และให้ไม่มี background หรือให้โปร่งใส และมีรูปแบบคำสั่งดังนี้

theSym.SetStyle(#RASTERFILL\_STYLE\_SOLID)

เราอาจจะตั้งค่าของสีโดยอ้างอิงจากข้างต้น และขนาดของเส้น outline ดังนี้

theSym.SetColor(colorRed)

theSym.SetOLWidth(3)

จากนั้นให้ทำการวาดภาพ graphic ใหม่อีกครั้งด้วยสัญลักษณ์ใหม่ด้วยคำสั่ง

theCircleGraphic.Invalidate

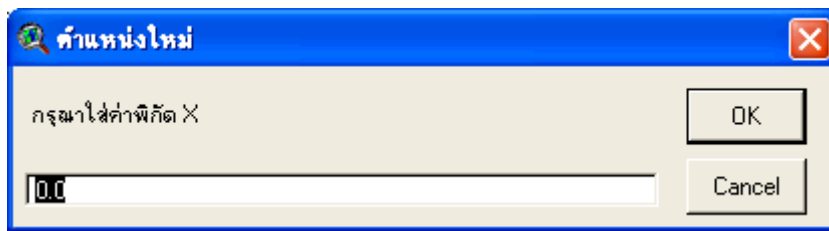
### การสร้าง Shape จากการ Key-in

เราอาจจะรอรับค่าจากผู้ใช้เป็นผู้กรอกพิกัดภูมิศาสตร์ โดยอาศัย Message Box เป็นเครื่องมือในการรับค่าพิกัดแล้วนำมาใช้ในการสร้างจุด shape ได้ โดยมีรูปแบบดังนี้

ในการรับค่าผ่าน Message Box โดยใช้คำสั่ง request คือ input

x = MsgBox.Input("กรุณาใส่ค่าพิกัด X", "ตำแหน่งใหม่", "0.0")

y = MsgBox.Input("กรุณาใส่ค่าพิกัด Y", "ตำแหน่งใหม่", "0.0")



จากนั้นเมื่อได้ค่าจาก MsgBox.Input ข้อมูลที่ได้รับผ่าน Message Box จะอยู่ในรูปแบบ String ดังนั้นหากเราต้องการเป็นข้อมูลตัวเลขเพื่อใช้ในการประมวลผลจะต้องทำการเปลี่ยนแปลงให้เป็นข้อมูล Number ดังนี้

```
xCoor = x.AsNumber
```

```
yCoor = y.AsNumber
```

จากนั้นเราจึงนำค่าพิกัดที่ได้ไปสร้าง point shape ได้

```
thePoint = Point.Make(xCoor, yCoor)
```

#### การตรวจสอบค่านำเข้า

ในการที่ผู้ใช้กรอกค่านำเข้าหรือไม่ได้กรอกข้อมูลใดๆ จะมีค่าเป็น nil หรืออาจจะกดปุ่ม Cancel นั้นให้เราทำการตรวจสอบ ก่อนนำค่าไปใช้งาน โดยใช้รูปแบบคำสั่งดังนี้

```
if (x = nil) then
```

```
MsgBox.Warning(“ยังไม่ได้ใส่ข้อมูลค่าพิกัด X ครบ”, “”)
```

```
Exit
```

```
End
```

หรือถ้าผู้ใช้กรอกค่าอื่นๆ ที่ไม่ใช่ตัวเลข ให้ทำการแจ้งให้ผู้ใช้ทราบ ดังนี้

```
if (x.IsNumber.Not) then
```

```
MsgBox.Warinig(“ข้อมูลที่ป้อนไม่ใช่ค่าตัวเลขครับ”, “”)
```

Exit

End

### การสร้าง Shape โดยการใช้ Mouse

ในการที่เราต้องการให้ผู้ใช้งานสร้าง Shape โดยการใช้ mouse คลิกเลือกไปบน View จะต้องใช้ request คือ `returnuser<type>` ดังนี้

ขั้นแรกจะต้องรับค่าขนาดพื้นที่ของ map display จาก View ก่อน

`theDisplay = theView.GetDisplay`

ถ้าต้องการสร้าง point shape ให้ใช้คำสั่ง

`thePoint = theDisplay.ReturnUserPoint`

ถ้าต้องการสร้าง circle shape ให้ใช้คำสั่ง

`theCircle = theDisplay.ReturnUserCircle`

ถ้าต้องการสร้าง rectangle shape ให้ใช้คำสั่ง

`theRect = theDisplay.ReturnUserRect`

จากนั้นผู้ใช้งานก็ทำการคลิกเมาส์ไปบนหน้าจอ View ได้เพื่อสร้าง shape

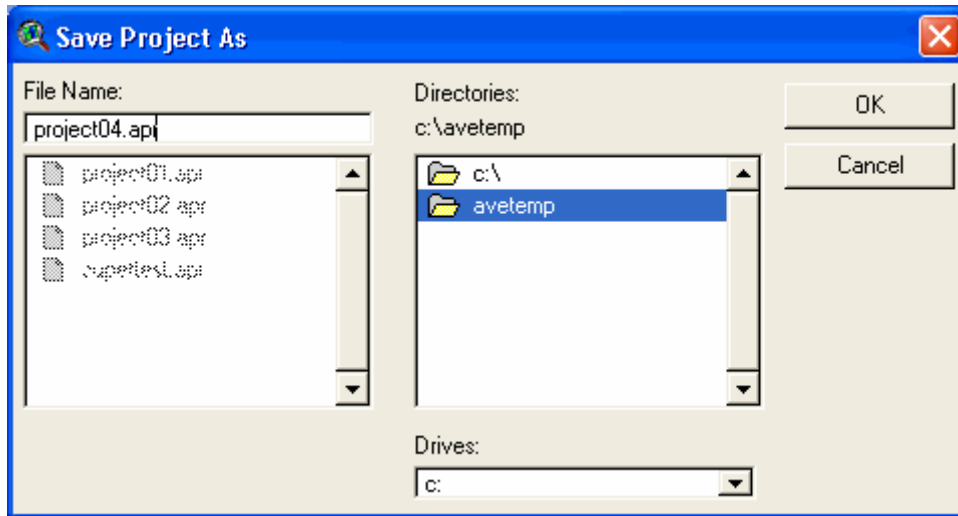
### แบบฝึกหัดที่ 7.1 การสร้าง Shape และ Graphic ประเภทจุด

ให้ทุกท่านเปิด Project ที่ทำไว้ในบทความครั้งที่ 6 ที่ผ่านมาแล้วให้ลองทดสอบ script ใหม่ เพื่อให้เราเข้าใจในบทเรียนมากขึ้น

**C:\AVTEMP\PROJECT03.APR**

จากนั้นให้ทำการบันทึกแฟ้มใหม่เป็นชื่อ **PROJECT04.APR** โดยเลือกไปที่คำสั่ง File - Save

Project As... แล้วเลือกไปยัง C:\AVTEMP ซึ่งเป็นฐานที่เก็บข้อมูลตัวอย่างโปรเจกต์ฝึกหัดของเรา



ให้เปิด Script Editor ขึ้นมาใหม่ โดยที่ไป double click ที่ปุ่ม script icon ที่ Project window



ให้เลือกไปที่เมนู Script - Properties... ให้เปลี่ยนชื่อ Script เป็น GUIDrawPoint แล้วกดปุ่ม OK

สมมติว่าต้องการสร้างจุดบนรูปแผนที่ประเทศไทย ซึ่งได้ไปทำการวัดพิกัดมาแล้วที่ (9377817.10317,1534527.217321) ดังนั้นจึงต้องทำการเลือกแผนที่เฉพาะดังนี้

เมื่อเราส่ง request ที่ชื่อ GetProject ไปให้กับวัตถุที่ชื่อ av มันจะทำการให้ค่าที่มีอยู่ในปัจจุบัน นั่นคือ "PROJECT04.AP" ไปยังตัวแปรที่ชื่อ theProject

**theProject = av.GetProject**

ให้สร้างตัวแปรที่ชื่อ theView เพื่อรับค่าจาก request คือ application เพื่อตรวจสอบว่ามี Document หรือ View ชื่อ World มีอยู่ในฐานข้อมูล โดยให้พิมพ์

**theView = theProject.FindDoc("World")**

ให้ View Document ที่พบเปิดขึ้นมาโดยให้ request ที่ชื่อ GetWin นั่นคือการรับ Document window และ request ที่สั่งให้ Open

**theView.GetWin.Open**

ให้สร้างตัวแปรชื่อ thePoint เพื่อรับค่าพิกัดของข้อมูลแบบจุด

**thePoint = Point.Make(9377817.10317,1534527.217321)**



เราเริ่มสร้างข้อมูล point shape ในรูปแบบ graphic โดยสร้างตัวแปรชื่อ thePointGraphic เพื่อส่งค่า Graphic ซึ่งมีข้อมูลพิกัดข้อมูลแบบจุด Shape

**thePointGraphic = GraphicShape.Make(thePoint)**

เราดัดแปลงสีและคุณสมบัติของ graphic ดังนี้

สร้างตัวแปร theSymbol เพื่อให้รับค่าจาก thePointGraphic ซึ่งจะรับค่าจากจุดที่สร้างขึ้น

**theSymbol = thePointGraphic.GetSymbol**

ให้ทำการตั้งค่าตัวแปร blue เพื่อรับค่าสีน้ำเงิน

**blue = Color.GetBlue**

ให้ตั้งค่าสีลงในตัวแปร theSymbol ที่ตั้งไว้เมื่อก่อนหน้านี้

**theSymbol.SetColor(blue)**

กำหนดขนาดของ theSymbol ได้

**theSymbol.SetSize(8)**

เราจะเพิ่มข้อมูลแบบจุดใน graphic list และวาดข้อมูล graphic ลงบน view

ตั้งค่าตัวแปรชื่อ theGraphicList เพื่อรับค่าจาก theView ที่ตรวจสอบหาข้อมูล Graphic ที่แสดงผล

**theGraphicList = theView.GetGraphics**

เพิ่มข้อมูล thePointGraphic ลงใน graphic list

```

theProject = av.GetProject
theView = theProject.FindDoc("World")
theView .GetWin.Open
thePoint = Point.Make(9377817.10317,1534527.217321)
thePointGraphic = GraphicShape.Make(thePoint)
theSymbol = thePointGraphic.GetSymbol
blue = Color.GetBlue
theSymbol.SetColor(blue)
theSymbol.SetSize(8)
theGraphicList = theView .GetGraphics
    
```

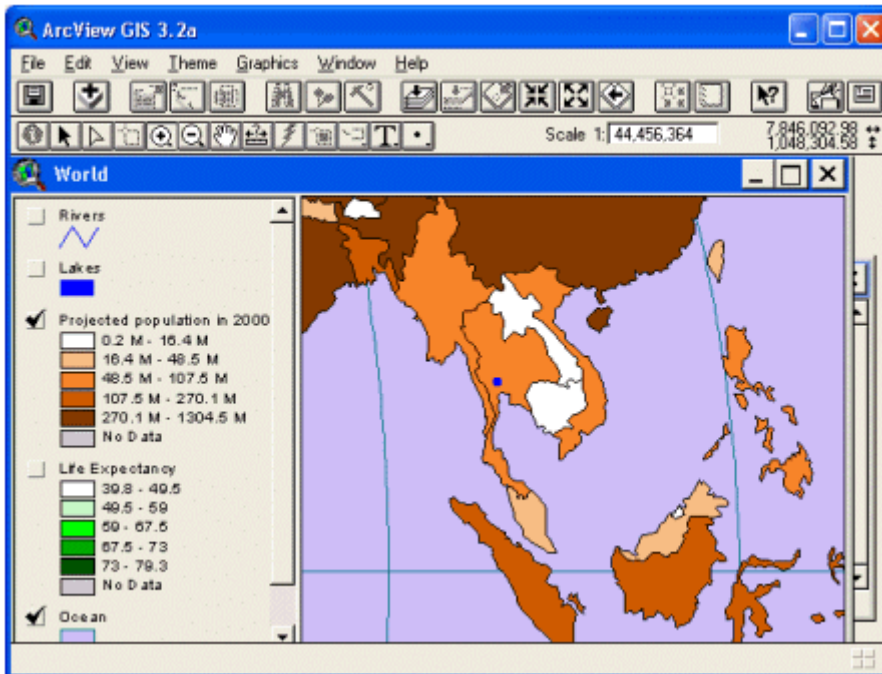
เมื่อพิมพ์ครบแล้วให้เราทำการ compile โปรแกรมนี้ด้วยคำสั่ง compile



ทำการทดสอบโปรแกรมโดยการประมวลผลดู



จะได้ผลดังนี้



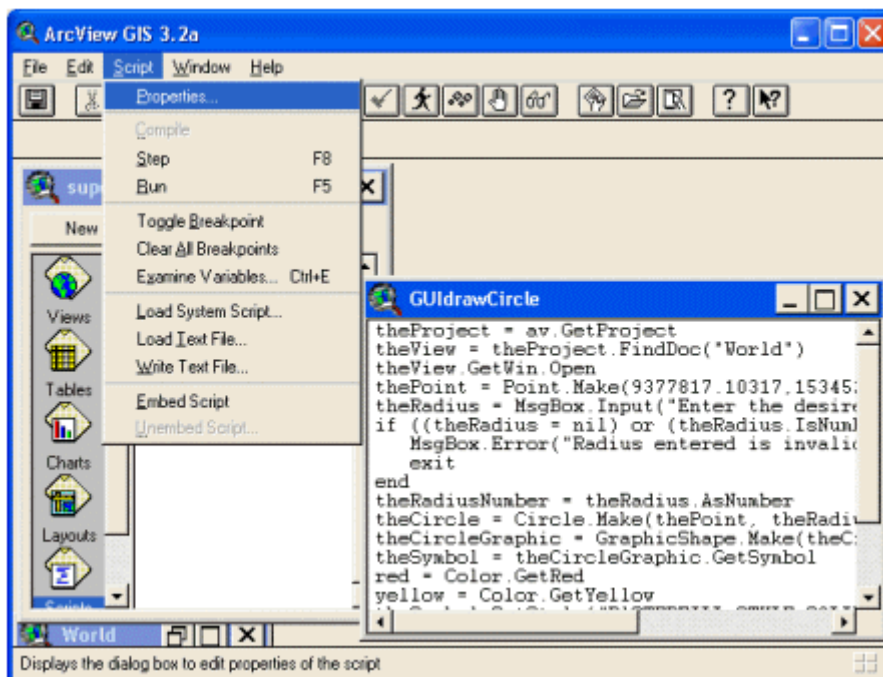
## แบบฝึกหัดที่ 7.2 การสร้าง Shape และ Graphic ประเภทวงกลม

จากโปรเจค PROJECT04.APR เราจะเพิ่มในส่วนของการทำงานวงกลมขึ้นมา เมื่อบางท่านอาจจะนำไปประยุกต์ใช้งานได้ในกรณีที่ต้องการ shade สีหรือทำจุดที่เป็น landmark ให้เด่น ดังนี้

ให้เปิด Script Editor ขึ้นมาใหม่ โดยที่ไป double click ที่ปุ่ม script icon ที่ Project window



ให้เลือกไปที่เมนู Script - Properties... ให้เปลี่ยนชื่อ Script เป็น GUIdrawCircle แล้วกดปุ่ม OK



เมื่อเราส่ง request ที่ชื่อ GetProject ไปให้กับวัตถุที่ชื่อ av มันจะทำการให้ค่าที่มีอยู่ในปัจจุบัน นั่นคือ "PROJECT04.AP" ไปยังตัวแปรที่ชื่อ theProject

**theProject = av.GetProject**

ให้สร้างตัวแปรที่ชื่อ theView เพื่อรับค่าจาก request คือ application เพื่อตรวจสอบว่ามี

Document หรือ View ชื่อ World มีอยู่ในฐานข้อมูล โดยให้พิมพ์

```
theView = theProject.FindDoc("World")
```

ให้ View Document ที่พบเปิดขึ้นมาโดยให้ request ที่ชื่อ GetWin นั่นคือการรับ Document window และ request ที่สั่งให้ Open

```
theView.GetWin.Open
```

ให้สร้างตัวแปรชื่อ thePoint เพื่อรับค่าพิกัดของข้อมูลจุดกึ่งกลางของวงกลม

```
thePoint = Point.Make(9377817.10317,1534527.217321)
```

ให้เราสร้างตัวรับค่าของรัศมีขึ้นมา โดยใช้ Message box เพื่อรับค่าผ่าน Input ดังนี้

```
theRadius = MsgBox.Input("Enter the desired radius","Circle Definition","0")
```

แต่อย่างไรก็ดีต้องมีการตรวจสอบค่าที่ได้รับ เพื่อมีความผิดพลาดในระหว่างการกรอกข้อมูลเกิดขึ้น คือ ถ้าไม่มีการป้อนค่าหรือเรียกว่า nil หรือเป็นข้อมูลที่ไม่ใช่ตัวเลข ให้เตือนขึ้นมาดังนี้

```
if ((theRadius = nil) or (theRadius.IsNumber.Not)) then
```

```
MsgBox.Error("Radius entered is invalid","No can do!")
```

```
exit
```

```
end
```

ถ้าเป็นการป้อนค่าที่ถูกต้อก็ให้ทำการรับค่า Radius ผ่านตัวแปร theRadiusNumber ดังนี้

```
theRadiusNumber = theRadius.AsNumber
```

ให้สร้างวงกลมขึ้นมาโดยมีตัวแปร theCircle รับค่าจากการสร้างวงกลมโดยอาศัยข้อมูลจากตัวแปร thePoint เป็นจุดกึ่งกลาง และ theRadiusNumber เป็นรัศมีของวงกลม

```
theCircle = Circle.Make(thePoint, theRadiusNumber)
```

สร้างตัวแปรขึ้นมารับค่าชื่อ theCircleGraphic เพื่อรับค่าการสร้างรูปภาพวงกลม

```
theCircleGraphic = GraphicShape.Make(theCircle)
```

กำหนดค่าสัญลักษณ์ของวงกลมดังนี้

```
theSymbol = theCircleGraphic.GetSymbol
```

```
red = Color.GetRed
```

```
yellow = Color.GetYellow
```

การกำหนดรูปแบบของ pattern

```
theSymbol.SetStyle(#RASTERFILL_STYLE_SOLID)
```

ตั้งค่าสีของ pattern

```
theSymbol.SetColor(yellow)
```

ตั้งค่าสีของเส้น outline

```
theSymbol.SetOLColor(red)
```

ตั้งค่าขนาดของเส้น outline

```
theSymbol.SetOLWidth(3)
```

ตั้งค่าตัวแปรชื่อ theGraphicList เพื่อรับค่าจาก theView ที่ตรวจสอบหาข้อมูล Graphic ที่แสดงผล

```
theGraphicList = theView.GetGraphics
```

เพิ่มข้อมูล theCircleGraphic ลงใน graphic list

```
theGraphicList.Add(theCircleGraphic)
```

```

theProject = av.GetProject
theView = theProject.FindDoc("World")
theView .GetWin.Open
thePoint = Point.Make(9377817.10317,1534527.217321)
theRadius = MsgBox.Input("Enter the desired radius","Circle Definition","0"
if ((theRadius = nil) or (theRadius.IsNumber.Not)) then
  MsgBox.Error("Radius entered is invalid","No can do!")
  exit
end
theRadiusNumber = theRadius.AsNumber
    
```

เมื่อพิมพ์ครบแล้วให้เราทำการ compile โปรแกรมนี้ด้วยคำสั่ง compile

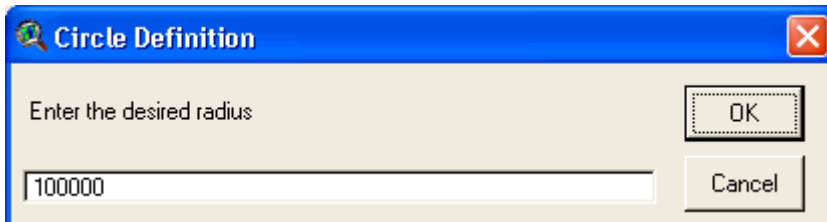


ทำการทดสอบโปรแกรมโดยการประมวลผลดู

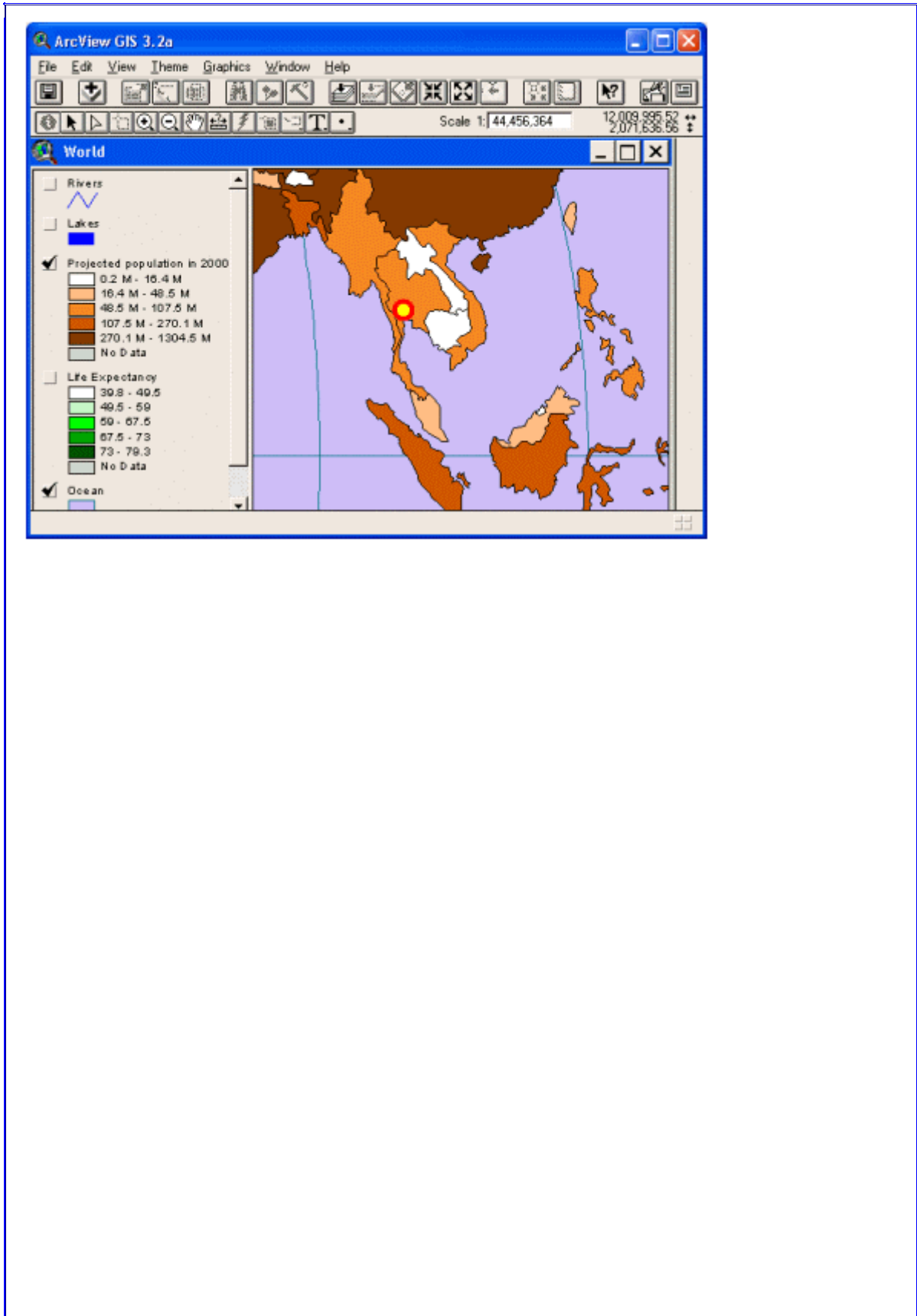


จะได้ผลดังนี้

ให้กำหนดค่ารัศมีเท่ากับ 100000 จะเห็นได้ชัดเจน

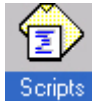


จะได้ผลดังรูป

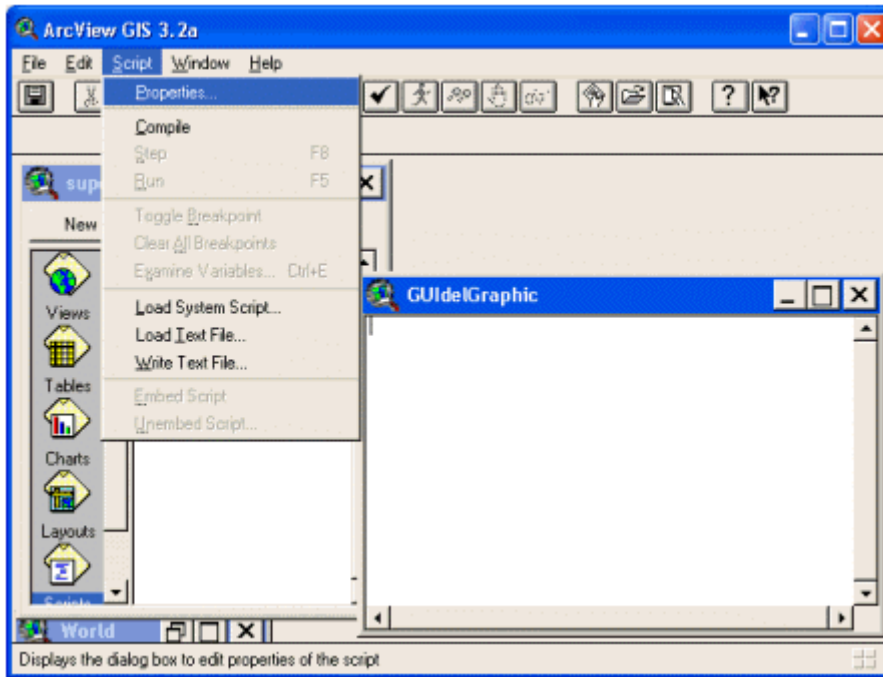


### แบบฝึกหัดที่ 7.3 การลบข้อมูล Graphic ทั้งหมด

จากโปรเจค PROJECT04.APR เราจะเพิ่มในส่วนของลบข้อมูล Graphic ที่ได้วาดขึ้นมา ดังนี้  
ให้เปิด Script Editor ขึ้นมาใหม่ โดยที่ไป double click ที่ปุ่ม script icon ที่ Project window



ให้เลือกไปที่เมนู Script - Properties... ให้เปลี่ยนชื่อ Script เป็น GUIDelGraphic แล้วกดปุ่ม OK

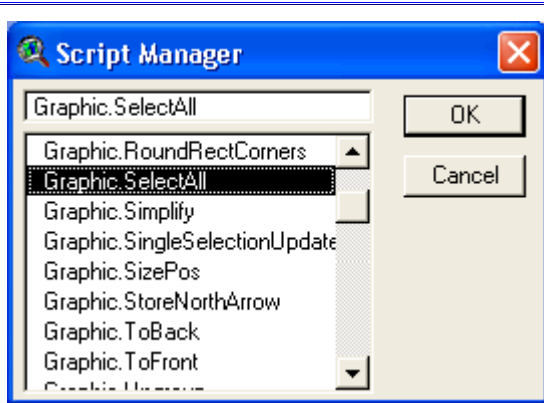


จากนั้นให้คลิกเลือกปุ่ม Load System Script เพื่อเรียกคำสั่งของระบบที่มีอยู่



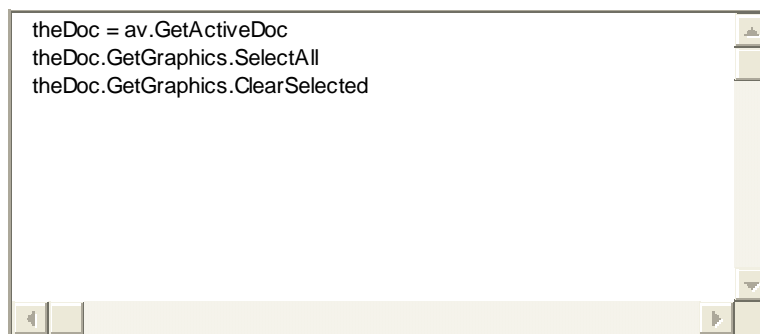
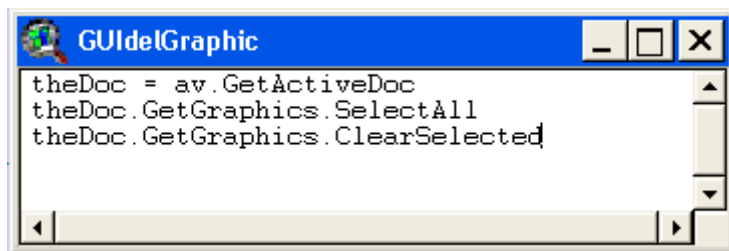
ให้เลือกชุดคำสั่งชื่อ Graphic.SelectAll แล้วกดปุ่ม OK





จากนั้นให้พิมพ์คำสั่งเพิ่มขึ้นมาเพื่อลบข้อมูล Graphic ดังนี้

`theDoc.GetGraphics.ClearSelected`



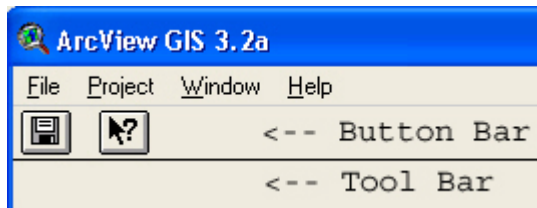
เมื่อพิมพ์ครบแล้วให้เราทำการ compile โปรแกรมนี้ด้วยคำสั่ง compile



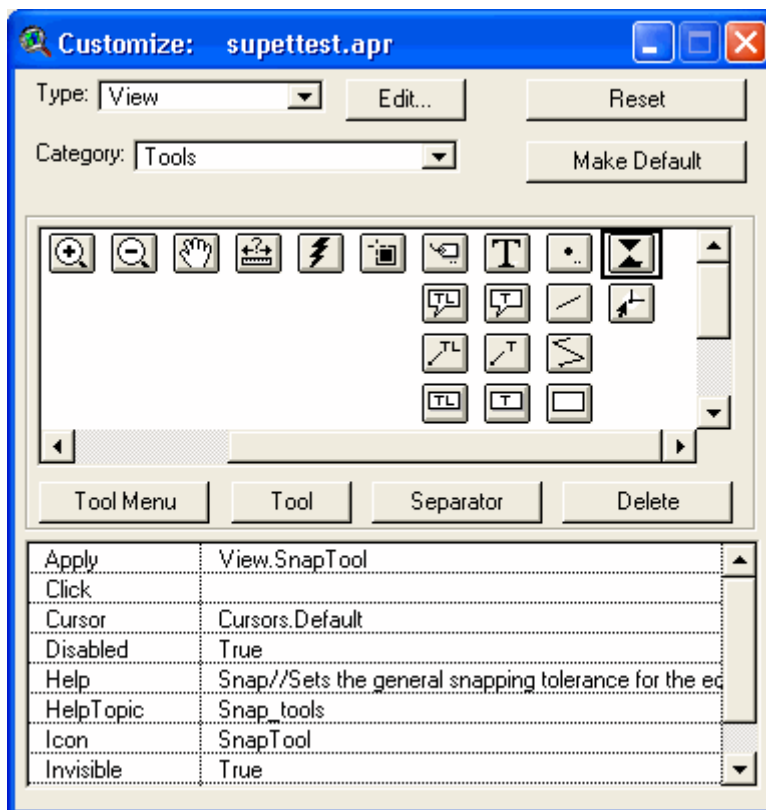
### แบบฝึกหัดที่ 7.4 การสร้าง Tool button

จากโปรเจค PROJECT04.APR เราจะเพิ่มในส่วนของคำสั่งจากแบบฝึกหัดที่ 7.1, 7.2, 7.3 เข้ามารวมไว้ในปุ่ม Tool เครื่องมือสำหรับวาดและลบ โดยดำเนินการดังนี้

ให้ Double Click ที่ Tool bar เพื่อเรียก Customize



ใน Customize : Dialog ให้เลือกที่แถบ Type: ให้เป็น View และ ในส่วนของ Category ให้เป็น Tools

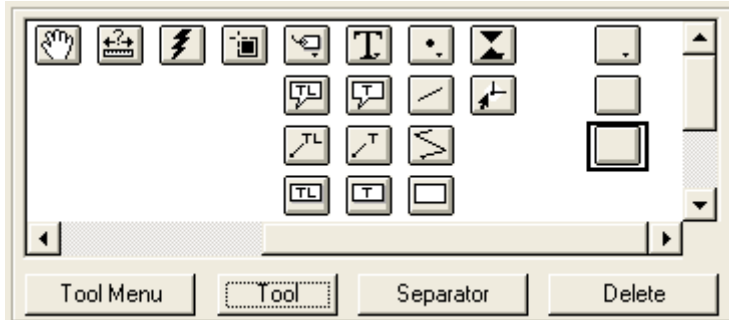


แล้วในส่วนของ Control Editor หรือรูปปุ่มนั้นให้เลื่อนไปท้ายสุด แล้วเลือกที่ปุ่มท้ายสุด

กดปุ่ม Separator 1 ครั้งเพื่อเว้นช่องว่าง

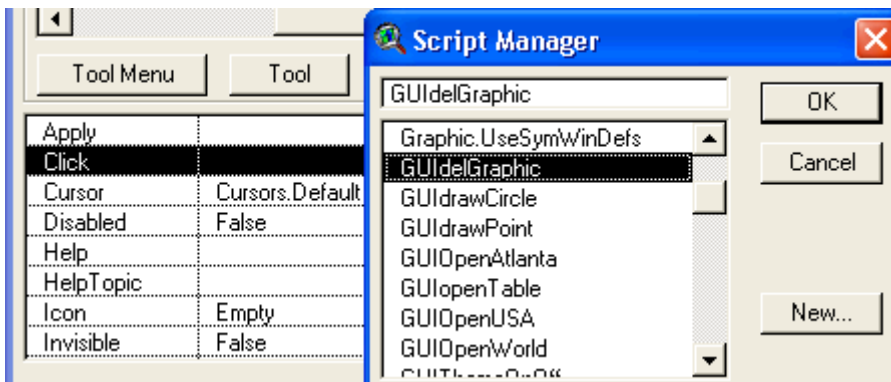
เลือกที่ปุ่ม Tool Menu 1 ครั้ง

และที่ปุ่ม Tool 2 ครั้งจะได้เมนู tool 3 ปุ่ม ดังรูป

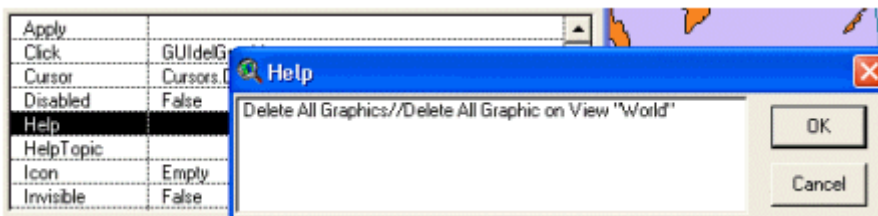


เลือกที่ปุ่มบนสุดปุ่มที่ 1 ให้เป็นตัวอย่างที่ 7.3 คือชุดคำสั่ง GUIdeGraphic โดยกำหนดดังนี้

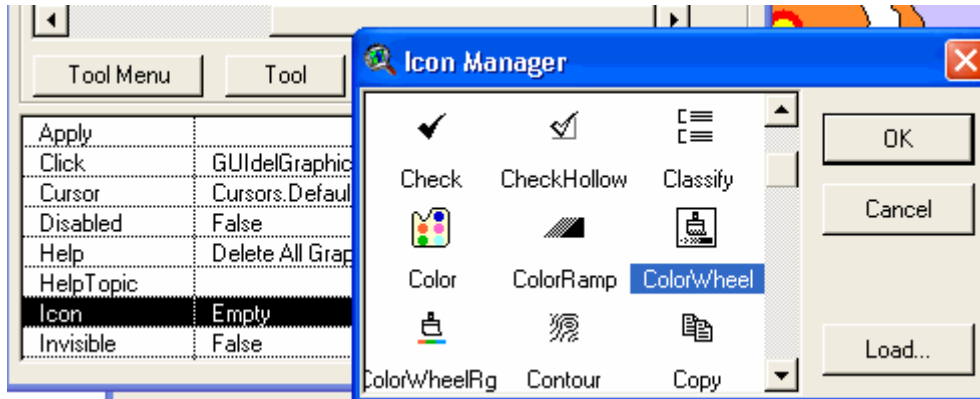
Double click ที่แถบชื่อ click ให้เลือกชุดคำสั่ง GUIdeGraphic



Double click ที่แถบชื่อ Help ให้พิมพ์ว่า Delete All Graphics//Delete All Graphic on View "World"



Double click ที่แถบชื่อ icon ให้เลือก ColorWheel



เลือกที่ปุ่มบนสุดปุ่มที่ 2 ให้เป็นตัวอย่างที่ 7.1 คือชุดคำสั่ง GUIDrawPoint โดยกำหนดดังนี้

Double click ที่แถบชื่อ click ให้เลือกชุดคำสั่ง GUIDrawPoint

Double click ที่แถบชื่อ Help ให้พิมพ์ว่า Draw Point//Draw Point on View "World"

Double click ที่แถบชื่อ icon ให้เลือก PennantBlue

เลือกที่ปุ่มบนสุดปุ่มที่ 3 ให้เป็นตัวอย่างที่ 7.2 คือชุดคำสั่ง GUIDrawCircle โดยกำหนดดังนี้

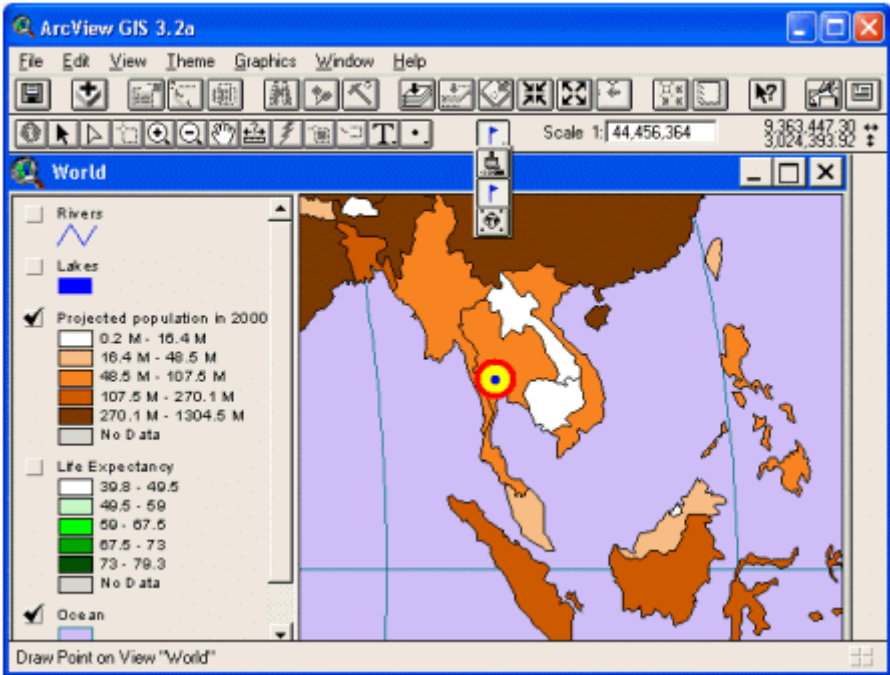
Double click ที่แถบชื่อ click ให้เลือกชุดคำสั่ง GUIDrawCircle

Double click ที่แถบชื่อ Help ให้พิมพ์ว่า Draw Circle//Draw Circle on View "World"


Double click ที่แถบชื่อ icon ให้เลือก FrameView

เมื่อกำหนดเสร็จให้ทำการปิดหน้าต่าง Customize... ออกมา

แล้วทำการทดสอบปุ่มคำสั่งต่างบน Tool button



เมื่อทดสอบเสร็จให้ทำการบันทึกโปรเจคโดยกดเมนู File - Save Project หรือ



ด้วยครับ เพื่อเก็บไว้ใช้ต่อไปในบทถัดไป