

Chapter 1: Introduction to Maya Embedded Language (MEL)

Topics:

Command line, command shell, script editor,
NURBS primitives, polygon primitives, and MEL
basic command

เอกสารประกอบการเรียน
รายวิชา ANI 951301
สาขาวิชาแอนิเมชันและเกม
วิทยาลัยศิลปะ สื่อ และ
เทคโนโลยี
มหาวิทยาลัยเชียงใหม่

Chapter 1: Introduction to Maya Embedded Language (MEL)

วัตถุประสงค์

1. เพื่อสร้างความรู้พื้นฐานและความเข้าใจให้นักศึกษาสามารถเริ่มต้นใช้ MEL Scripting Language ได้
2. เข้าใจถึงการทำงานของ MEL User Interface
3. สามารถเริ่มต้นสร้างรูปทรงพื้นฐานด้วย MEL และสามารถจัดการกับรูปทรงพื้นฐานได้
4. สามารถใช้คำสั่งเบื้องต้นเพื่อเป็นแนวทางสู่การประยุกต์ใช้ในเบื้องต้นต่อไป

เนื้อหาการสอน

1. การใช้ command line, command shell, และ script editor
2. การสร้าง NURBS primitives และ polygon primitives
3. คำสั่งการทำงาน MEL scripts ที่จำเป็นเบื้องต้น

An Introduction to MEL Scripting Language

Maya Embedded Language (MEL) เป็นส่วนสำคัญในการใช้ Maya ให้มีประสิทธิภาพสูงสุด แต่ผู้ใช้ส่วนใหญ่ยังมองข้ามการใช้งาน function ตัวนี้ ด้วยความกลัวที่ว่าจะต้องเขียนโปรแกรมหรือกลัวว่าจะต้องเรียนรู้สิ่งใหม่ๆ ซึ่งเป็นความเข้าใจที่ไม่ถูกต้องร้อยเปอร์เซ็นต์ การใช้ MEL ให้เกิดประโยชน์เฉพาะทางของทางแอนิเมชัน ผู้ใช้ไม่จำเป็นต้องมีความสามารถในการเขียนโปรแกรมขั้นสูง เพียงแต่ต้องมีความเข้าใจว่าเราควรจะใช้ MEL เพื่อตอบโจทย์ข้อไหน ในสถานการณ์ไหนมากกว่า

MEL ไม่ใช่เป็นเพียงแค่องค์ประกอบหนึ่งของ Maya แต่เป็นรากฐานสำคัญที่ทำให้ Maya สามารถปฏิบัติการได้ function ต่างๆของ Maya ที่เราเรียกใช้จาก interfaces ของโปรแกรม มีการ run scripts อยู่เบื้องหลังแทบจะทั้งหมด ผู้ใช้ที่เข้าใจ MEL สามารถ customising workspace, ดัดแปลง tools ที่มีอยู่แล้ว หรือแม้กระทั่งสร้าง tools ตัวใหม่ขึ้นมาใช้เองให้ตอบสนองความต้องการในการทำงานของตน

การเขียนโปรแกรมด้วย MEL ก็เหมือนกับการเขียนโปรแกรมด้วย programming languages อื่นๆ เหมือนกับการเขียนภาษาอังกฤษที่เราใช้ในชีวิตประจำวันคือเราต้องเข้าใจ vocabulary และกฎการใช้ grammar ของภาษาที่จะใช้ ดังนั้นเราจึงต้องทำความเข้าใจความหมายของคำสั่งแต่ละตัวและวิธีผูกโยงคำสั่งเหล่านั้นให้เกิดความหมาย

เป้าหมายของการเรียนการสอนวิชานี้ มุ่งเน้นให้นักศึกษาเข้าใจใน syntax, vocabulary และ structure ของ MEL เพื่อไปประยุกต์ใช้ให้เกิดประโยชน์สูงสุดในการพัฒนาการสร้างสรรค์แอนิเมชันขั้นสูงต่อไป โดยจะพยายามหลีกเลี่ยงการใช้ math ขั้นสูง แต่จะพยายามใช้เพียง basic math เพื่อความง่ายต่อการเข้าใจ และหวังว่านักศึกษาที่มีความสนใจจะนำ concept ไปต่อยอดด้วยตนเองเมื่อต้องทำงานจริงต่อไป

MEL Basics

ในหัวข้อต่อไปเราจะมาทำความเข้าใจพื้นฐานความรู้ที่จำเป็นก่อนการใช้งาน MEL หน้าต่างการทำงานและองค์ประกอบ interface ของ MEL ดังที่กล่าวไว้ข้างต้นว่า MEL คือสิ่งที่ทำงานอยู่เบื้องหลัง interface อันสวยงามของ Maya สิ่งที่สร้างหน้าต่างๆ interface ต่างๆภายในโปรแกรมก็คือ MEL นั่นเอง เราลองมาดูว่าเราจะสามารถเรียกใช้การทำงานของ MEL ได้อย่างไร

Maya UI to Work with MEL (Interface)

จากหน้าต่างการทำงานของโปรแกรม Maya เราสามารถเข้าถึงการใช้คำสั่ง MEL ได้สามทางคือผ่านทาง command line, command shell และ script editor

1. The Command Line

การเรียกใช้ MEL ผ่าน command line เป็นวิธีที่สะดวกที่สุด เนื่องจากผู้ใช้สามารถพิมพ์คำสั่งได้ทันทีที่หน้าต่างด้านล่างของโปรแกรมดังภาพประกอบด้านล่าง

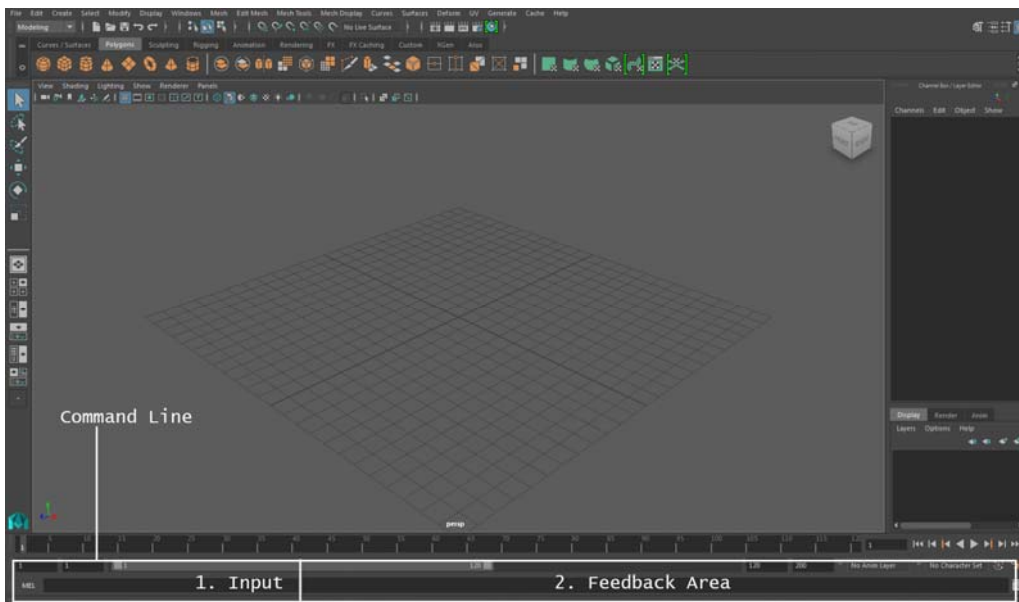


Fig 01-01: แสดงตำแหน่งของ command line บนหน้าต่างการทำงานของโปรแกรม Maya

หน้าต่าง command line ประกอบด้วยสองส่วนหลักคือส่วนของ commands input ที่เราสามารถพิมพ์ตัวคำสั่งลงไปได้ และส่วน feedback area ที่โปรแกรมจะแสดงผลของการ run คำสั่งที่เราป้อนเข้าไป ชุดคำสั่งที่เหมาะสมกับการใช้ในขณะนี้ ควรเป็นคำสั่งสั้นๆ ที่ใช้ประกอบการทำงานปกติ เช่นคำสั่งเรียก primitive shapes หรือคำสั่ง rotation เนื่องจากมีเนื้อที่ในการป้อนคำสั่งเพียงบรรทัดเดียว จึงไม่สามารถป้อนคำสั่งที่มีความซับซ้อนสูงได้

ให้นักศึกษาทดลองพิมพ์คำสั่ง polySphere ลงในช่อง input ของ command line จากนั้นกด enter จะปรากฏ polygon sphere ขึ้นที่หน้าต่าง workspace ที่กำลังทำงานอยู่ เนื่องจาก polySphere คือคำสั่งในการสร้าง polygon sphere นั่นเอง และที่หน้าต่าง feedback area จะปรากฏข้อความว่า // Result: pSphere1 polySphere1 แสดงว่ามี polySphere ถูกสร้างขึ้นมา

เราจะสังเกตได้ว่าทุกครั้งที่เรากด enter โปรแกรมจะทำการ run ชุดคำสั่งที่ป้อนไว้ และชุดคำสั่งที่ป้อนไว้จะหายไปจากช่อง input โดยที่ผลลัพธ์ของการ run คำสั่งแสดงในช่อง feedback area ถ้าคำสั่งที่เราป้อนมีข้อผิดพลาด จะมีข้อความ // Error: ตามด้วยสาเหตุที่ผิดพลาดแสดงให้เราดูทางด้าน feedback area

จากนั้นให้นักศึกษาทดลองพิมพ์คำสั่ง `scale -y 3` ลงบนช่อง input จากนั้น enter เราจะพบว่า sphere ที่เราสร้างถูกขยายออกสามเท่าตามแนวแกน Y นี่คือนิยามคร่าวๆถึงพื้นฐานการทำงานในส่วนของ command line ซึ่งจะกล่าวถึงในเบื้องต้นต่อไป

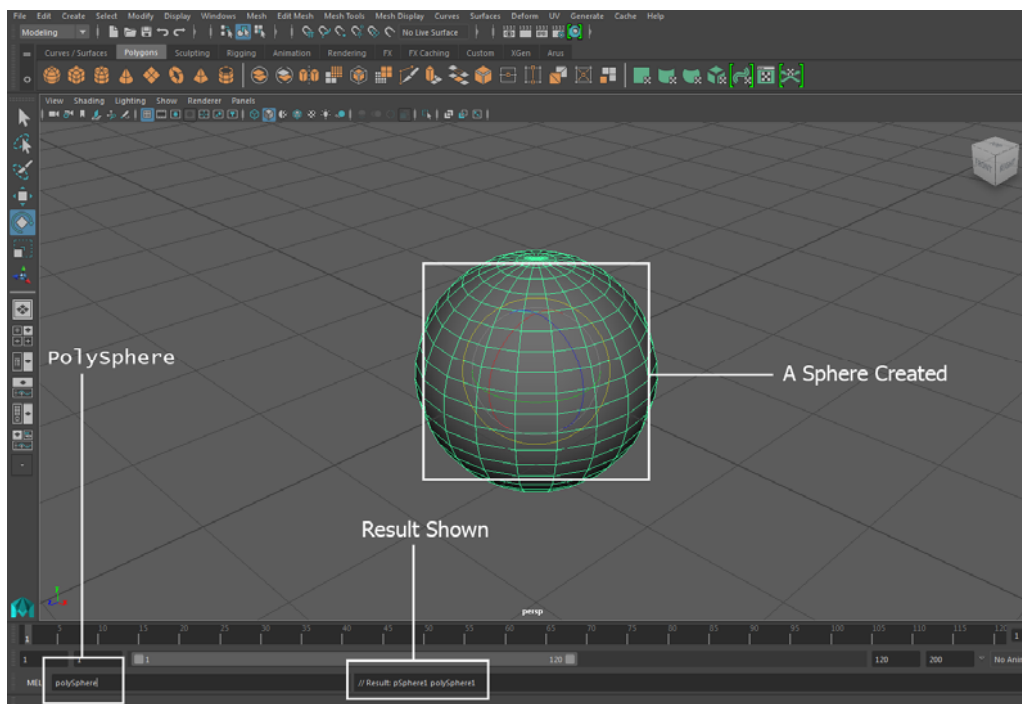


Fig 01-02: ตัวอย่างการใช้คำสั่ง polySphere ในการสร้างรูปทรงกลม

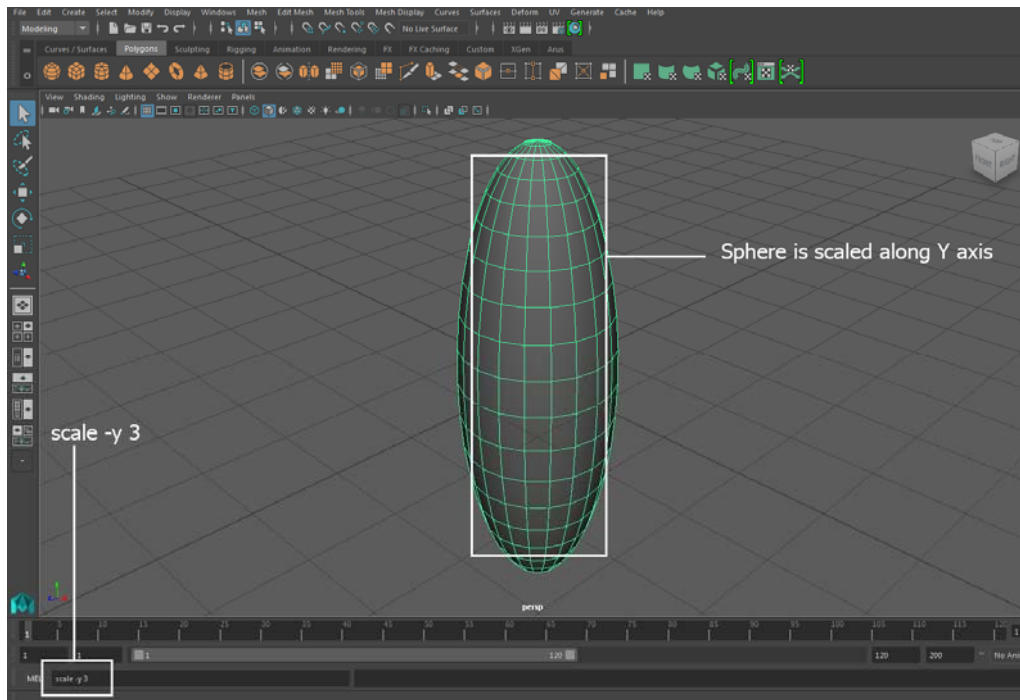


Fig 01-03: ตัวอย่างการใช้คำสั่ง scale -y 3 เพื่อปรับสัดส่วนของรูปทรง

2. The Command Shell

หน้าต่าง command shell สามารถเข้าถึงได้โดยการไปที่ Window>General Editors>Command Shell ลักษณะพิเศษของ command shell คือจะรวมการแสดงผลของทั้ง command line และ feedback ไว้ในหน้าต่างเดียวกัน หน้าต่าง command shell มีลักษณะการเรียกใช้คล้ายกับการเรียกคำสั่งในระบบปฏิบัติการ Unix Terminal หรือ DOS prompt มีประโยชน์คือสามารถตรวจสอบถึงชุดคำสั่งที่ run ก่อนหน้าได้อย่างสะดวก แต่เนื่องจากการรวม command line และ feedback ไว้ด้วยกันทำให้กินพื้นที่และอาจไม่เหมาะสมในการป้อนคำสั่งที่มีความซับซ้อนสูง หรือต้องการพื้นที่ในการทำงานมาก

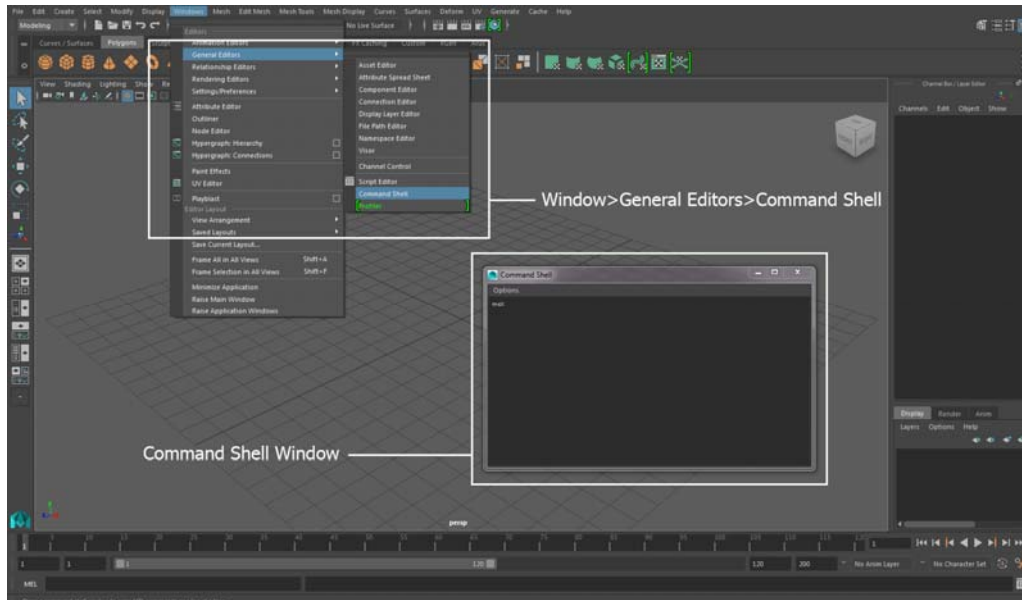


Fig 01-04: แสดงตำแหน่งของ command shell บนหน้าต่างการทำงานของโปรแกรม Maya

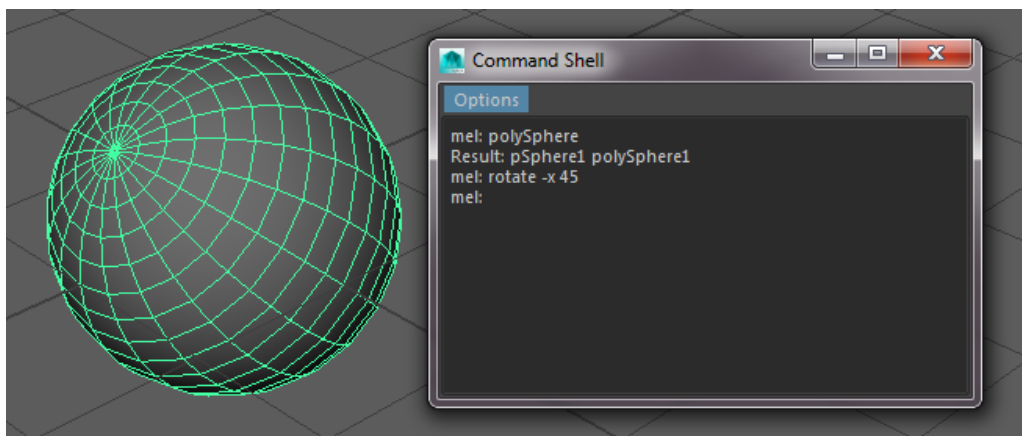


Fig 01-05: ตัวอย่างหน้าต่าง command shell เมื่อเรียกคำสั่งสร้าง sphere แล้วทำการหมุนตามแนวแกน X ไป 45

องศา

3. The Script Editor

หน้าต่าง script editor จะเป็นหน้าต่างหลักที่เราใช้ในการทำงานในภาคการศึกษาี้ โดยเราสามารถเข้าถึงหน้าต่าง script editor ได้สองวิธี วิธีแรกโดยไปที่ Window>General Editors>Script Editor หรือสามารถเข้าถึงได้ด้วยวิธีที่สองโดยการคลิกที่ icon ของ script editor ที่ด้านล่างขวามือถัดจาก feedback area ของ command line

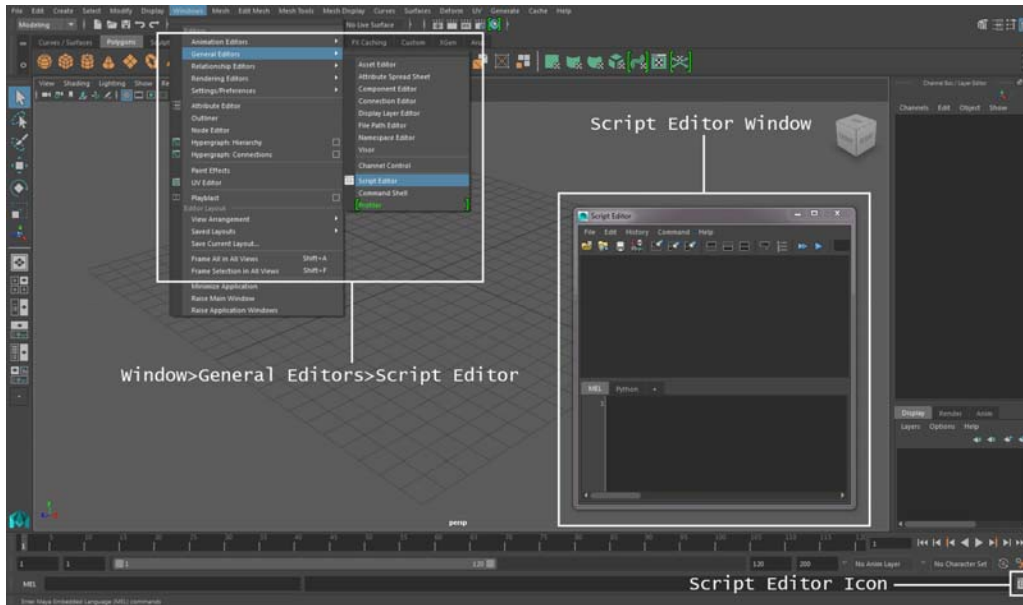


Fig 01-06: แสดงตำแหน่งของ script editor บนหน้าต่างการทำงานของโปรแกรม Maya

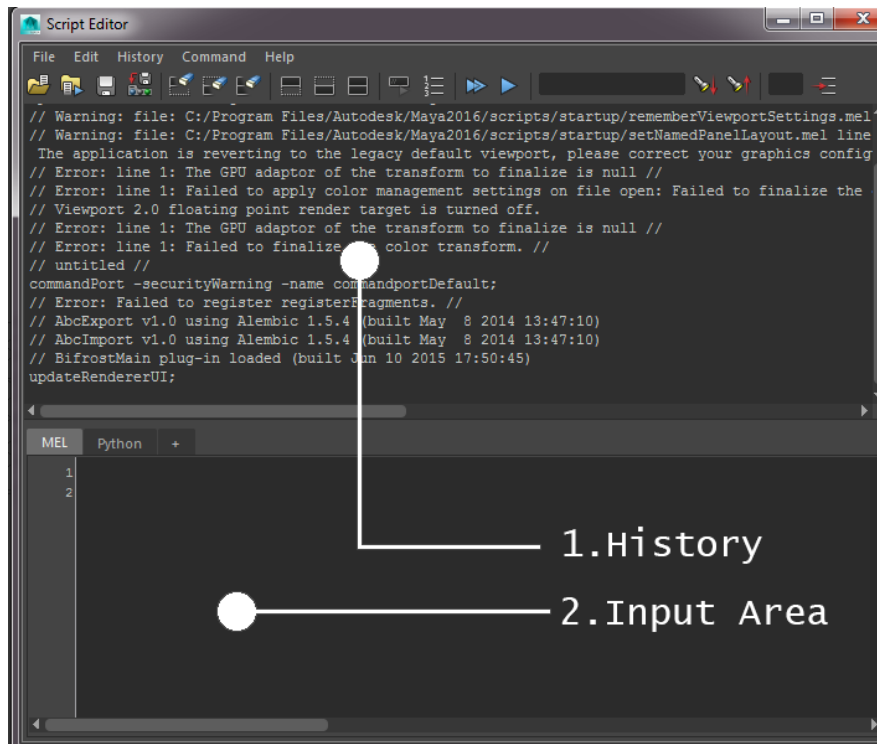


Fig 01-07: หน้าต่าง script editor ถูกแบ่งออกเป็นสองส่วนคือ 1) history และ 2) input

ทุกคำสั่งที่เราใช้ภายในโปรแกรม Maya จะถูกแสดงในส่วนของ history นี่เป็นวิธีที่ดีที่ที่เราจะสามารถเรียนรู้คำสั่งต่างๆภายในโปรแกรมว่ามีการเรียก scripts คำสั่งอย่างไร ยกตัวอย่างเช่นถ้าเรากดสร้าง polygon cube ด้วยวิธีปกติผ่านทาง Create>Polygon Primitives>Cube ที่หน้าต่าง history ของ script editor จะแสดงผลว่า

```
polyCube -w 1 -h 1 -d 1 -sx 1 -sy 1 -sz 1 -ax 0 1 0 -cuv 4
-ch 1;
```

ทำให้เรารู้ว่า polyCube คือคำสั่งในการสร้าง polygon cube โดยที่ flags ต่างๆเช่น -w, -h, -d, -sx, -sy, -sz, -ax, -cuv, -ch เป็นตัวกำหนดคุณลักษณะเฉพาะต่างๆของ cube ที่สร้างขึ้นมานั่นเอง

ที่แถบ history และ command ด้านบน เราสามารถกำหนดรายละเอียดการแสดงผลต่างๆเช่น show line numbers โปรแกรมจะแสดงตัวเลขบรรทัดของ input scripts เป็นต้น ซึ่งมีประโยชน์ในการทำงานและการ debugging ตัว scripts

ข้อดีในการทำงานด้วย script editor คือมีพื้นที่ในการทำงานมาก เนื่องจากมีการแบ่งส่วนของ input แยกจาก history อย่างชัดเจน และสามารถขยายหน้าต่างให้มีขนาดเหมาะสมกับการทำงานได้ตามต้องการ

ในการพิมพ์ชุดคำสั่งที่มีมากกว่าหนึ่งบรรทัดใน script editor อาจมีความแตกต่างจากการใช้งานการเขียนโปรแกรมอื่นบ้างเล็กน้อย นั่นคือผู้ใช้สามารถใช้ได้เพียงปุ่ม enter หลักในส่วนของแป้นพิมพ์ตัวหนังสือในการเพิ่มบรรทัด โดยไม่สามารถใช้ปุ่ม enter ในส่วนของ numeric keypad ได้ ในส่วนของการ execute คำสั่ง เราสามารถทำได้สามวิธีคือ โดยการกด icon ที่ด้านบนของหน้าต่าง script editor, การกด CTRL + ปุ่ม enter หลัก หรือใช้ปุ่ม enter ในส่วนของ numeric keypad ก็สามารทำได้ ในส่วนของ icons ที่มีความสำคัญต่างๆในการใช้งานมีดังนี้



Clear History: ลบข้อมูลทั้งหมดที่แสดงอยู่ในหน้าต่าง history



Clear Input: ลบข้อมูลทั้งหมดที่อยู่ในหน้าต่าง input



Clear All: ลบข้อมูลทั้งหมดภายในหน้าต่าง history และ input พร้อมกัน



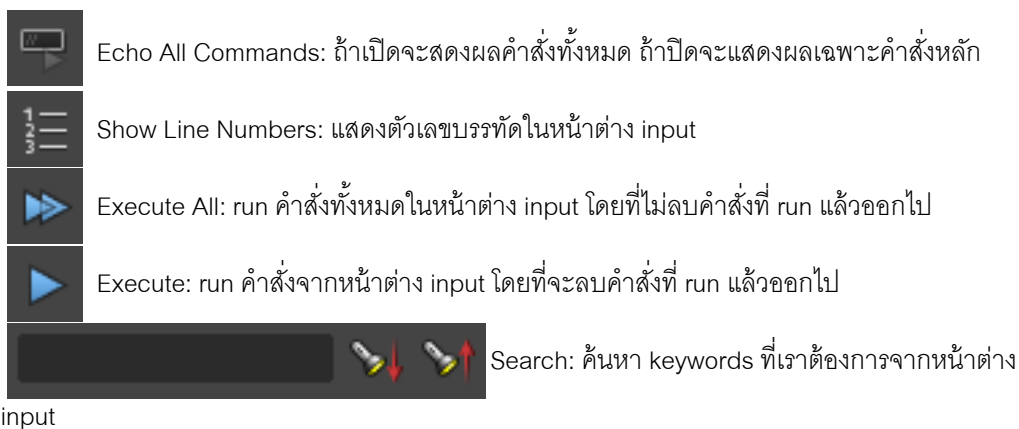
Show History: แสดงผลเฉพาะส่วนของหน้าต่าง history เท่านั้น



Show Input: แสดงผลเฉพาะส่วนของหน้าต่าง input เท่านั้น



Show Both: แสดงผลรวมทั้งในส่วนของหน้าต่าง history และ input พร้อมกัน



Basic Geometry

เมื่อนักศึกษามีความเข้าใจในหน้าต่างการทำงานแล้ว ต่อไปเราจะเริ่มทำงานด้วย MEL โดยการสร้างวัตถุขั้นพื้นฐาน เพื่อให้ให้นักศึกษาเกิดความคุ้นเคยกับการใช้งานโปรแกรม ที่นี่เรามาเรียนรู้คำสั่งในการสร้างรูปทรงแบบต่างๆกัน โดยจะแบ่งวัตถุออกเป็นสองประเภทหลักๆคือ NURBS และ Polygon โปรดจำไว้เสมอว่าใน MEL ทุกคำสั่งจะเป็นแบบ case-sensitive คือการรับค่าระหว่างตัวอักษรตัวเล็กและตัวใหญ่จะมีความแตกต่างกันนั่นเอง

NURBS Primitives

Sphere ใช้เรียกสร้างทรงกลม

nurbsCube ใช้เรียกสร้างลูกบาศก์

cylinder ใช้เรียกสร้างรูปทรงกระบอก

cone ใช้เรียกสร้างรูปทรงกรวย

nurbsPlane ใช้เรียกสร้างพื้นผิวระนาบ

torus ใช้เรียกสร้างรูปทรงวงแหวน

Polygon Primitives

polySphere ใช้เรียกสร้างทรงกลม

polyCube ใช้เรียกสร้างลูกบาศก์

polyCylinder ใช้เรียกสร้างรูปทรงกระบอก

polyCone ใช้เรียกสร้างรูปทรงกรวย

polyPlane ใช้เรียกสร้างพื้นผิวระนาบ

polyTorus ใช้เรียกสร้างรูปทรงวงแหวน

เมื่อเราใช้คำสั่งสร้างรูปทรงเหล่านี้ Maya จะสร้างวัตถุตามคุณลักษณะพื้นฐาน (default) ของวัตถุนั้นๆ ถ้าเรามีความต้องการกำหนดค่าต่างๆเพิ่มเติมเข้าไป สามารถทำได้โดยการกำหนด flag ควบคู่ไปกับคำสั่งสร้างวัตถุนั้นๆ flag ที่ใช้กำหนดค่าต่างๆภายใน MEL จะต้องนำหน้าด้วย เครื่องหมาย "-" เสมอ

ให้นักศึกษาลองสร้าง polygon cube โดยกำหนด flags ตามตัวอย่างข้างล่าง

```
polyCube -w 1 -h 3 -d 1 -sx 2 -sy 4 -sz 1 -ax 0 1 0;
```

เราจะได้ cube ที่มีขนาด กว้าง 1 unit สูง 3 units ลึก 2 units มี subdivisions ตามแนวแกน X=2, ตามแนวแกน Y=4 และตามแนวแกน Z=1 โดยที่จะถูกสร้างขึ้นตามแนวแกน Y

โดยนักศึกษสามารถตรวจสอบ flags ทั้งหมดที่ใช้ใน MEL ได้จากลิงค์ด้านล่าง

<http://help.autodesk.com/view/MAYAUL/2016/ENU/>

โดยให้พิมพ์ชื่อคำสั่งในการสร้างรูปทรงนั้นๆในช่อง search แล้วคุณลักษณะทุกอย่างของรูปทรงนั้นจะถูกแสดงออกมา

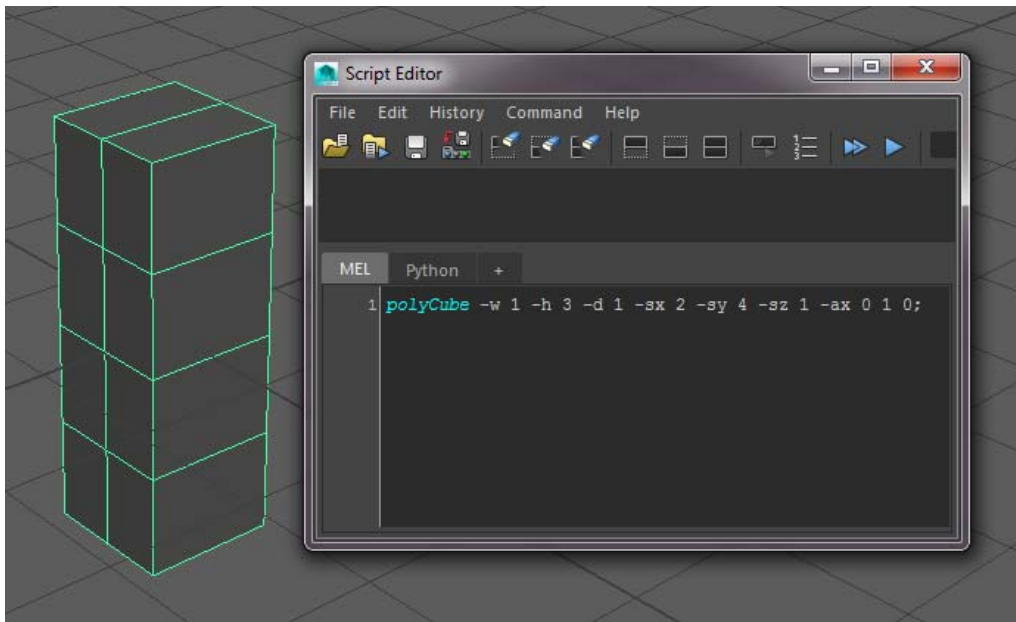


Fig 01-08: ตัวอย่างการใช้ flag ในการกำหนดคุณลักษณะให้กับวัตถุที่สร้างขึ้น

ให้นักศึกษาทดลองใช้คำสั่งอื่นๆกับรูปทรงเช่น move, rotate หรือ scale รวมทั้งทดลองใส่ flag ให้กับคำสั่งนั้นๆด้วย เมื่อเกิดความคุ้นเคยแล้วให้ทดลองสร้างภาพดังตัวอย่างข้างล่างโดยใช้ MEL scripts ในการทำทั้งหมด

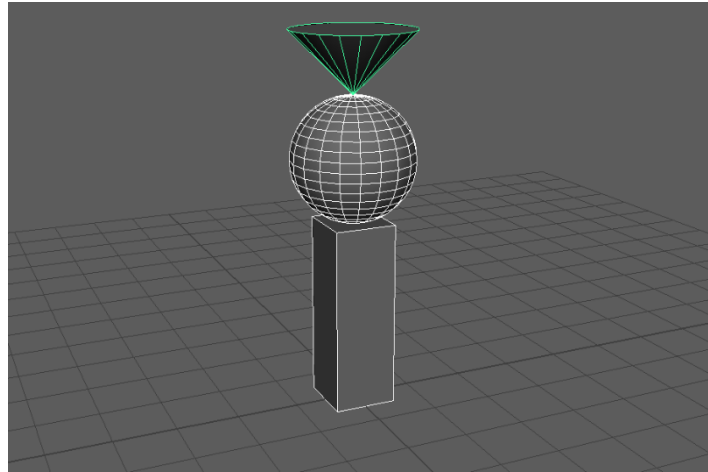


Fig 01-09: ภาพตัวอย่างเพื่อทดลองใช้คำสั่งและ flags ในการจัดวางและสร้างรูปทรงต่างๆ

โดยใจที่ยกกำหนดให้ สร้าง cube ขนาด 1x3x1 วางไว้บน grid lines แล้วสร้าง sphere ขนาดรัศมี 1 วางไว้ข้างบน cube แล้วจึงสร้าง cone กลับหัวขนาดรัศมีที่ฐาน = 1 และมีความสูง = 1 วางไว้บนทรงกลมอีกที

Flags ที่ควรต้องใช้คือ -w ความกว้าง, -h ความสูง, -d ความลึก, -r รัศมี

ทุกครั้งทีสิ้นสุดคำสั่งหนึ่งๆ ต้องปิดประโยคด้วยเครื่องหมาย “;” ด้วย

```
// วิธีที่สามารถทำได้
polyCube -w 1 -h 3 -d 1;
move -y 1.5;
polySphere -r 1;
move -y 4;
polyCone -r 1 -h 1;
rotate -x 180;
move -y 5.5;
```

Saving the Scripts to the Shelf

ในบางการทำงานเราอาจต้องเรียกใช้ scripts เดิมๆซ้ำๆ เราอาจต้องการสร้างให้เป็นแถบเครื่องมือเฉพาะของเราเอง เพื่อความสะดวกในการเรียกใช้ต่อไป ซึ่งสามารถทำได้ โดยการ select เลือกคำสั่งทั้งหมดที่เราต้องการจาก script editor จากนั้นใช้ mouse ปุ่มกลาง ลากข้อความ scripts นั้นๆมาวางไว้บน shelf ของ Maya ซึ่งจะปรากฏเป็น icon ใหม่ บน shelf ขึ้นมา เราสามารถเรียกใช้ได้โดยการกดที่ icon นั้น โดยเราสามารถกำหนดรูปของ icon ได้ รวมถึงข้อความอธิบาย และตัว scripts เองก็สามารถแก้ไขได้เช่นกัน

ให้นักศึกษาทดลองทำแถบเครื่องมือเพื่อใช้ในการลบทุกอย่างใน workspace โดยพิมพ์คำสั่ง

```
select -all;
doDelete;
```

จากนั้นให้สร้าง shelf ขึ้นมาเพื่อบรรจุคำสั่งนี้

เมื่อถึงส่วนนี้แล้วคาดว่านักศึกษาจะเริ่มมีความเข้าใจในการทำงานพื้นฐานของ MEL ได้ดีขึ้น ในส่วนต่อไปจะกล่าวถึงคำสั่งพื้นฐานที่จำเป็นสำหรับการทำงานเบื้องต้น เพื่อเตรียมความพร้อมให้นักศึกษาสามารถศึกษาเนื้อหาเบื้องต้นได้แบบต่อไป

Basic Commands

1. เราสามารถเรียกหน้าต่าง script editor ได้ง่ายอีกวิธีหนึ่งโดยการป้อนคำสั่ง ScriptEditor ที่ช่อง command line

2. เราสามารถย่อขยายหน้าต่าง script editor ได้ง่ายโดยการ CTRL + scroll mouse ปุ่มกลาง

3. ด้วยคำสั่ง print เราสามารถสั่งให้ Maya แสดงข้อความต่างๆที่ช่อง feedback area ได้ ให้นักศึกษาทดลองพิมพ์คำสั่ง

```
print "ชื่อตนเอง";
```

4. นอกจากพิมพ์คำสั่งตามที่บอกแล้ว เรายังสามารถใช้ print ในการคำนวณตัวเลขได้อีกด้วย แต่เราจะต้องใส่ตัวเลขไว้ในวงเล็บก่อน ให้นักศึกษาทดลองพิมพ์คำสั่ง

```
print (10/2);
```

เราจะได้ผลลัพธ์ว่า 5 ที่ช่อง feedback area

5. คำสั่ง print ยังมีประโยชน์อีกมากเพื่อใช้แสดงค่าต่างๆในการใช้ MEL ที่ซับซ้อนขึ้นในบทต่อไป

6. เราสามารถตั้งชื่อของ object ที่สร้างขึ้นมาได้ทันทีโดยไม่ต้องไปเปลี่ยนชื่อมันทีหลัง โดยใช้ flag "-name" ตามด้วยชื่อที่ต้องการ เช่นถ้าเราต้องการสร้าง cube ขึ้นมาโดยตั้งชื่อว่า Apple สามารถทำได้โดยใช้คำสั่ง

```
polyCube -name Apple;
```

7. แต่ถ้าเราต้องการแก้ไขชื่อวัตถุที่สร้างไว้แล้วสามารถทำได้ด้วยคำสั่ง rename

```
rename "ชื่อเก่าของมัน" "ชื่อใหม่ที่ต้องการ"
```

8. เราสามารถสั่งให้โปรแกรม select วัตถุได้ด้วยคำสั่ง `select -r` ชื่อวัตถุ เช่นถ้าเราจะเคลื่อนย้ายวัตถุชื่อว่า Apple ไปตามแนวแกน X เป็นระยะทาง 5 units สามารถทำได้โดย

```
select -r Apple;
move -x 5;
```

9. ถ้าต้องการเลือกวัตถุหลายๆชิ้นเพื่อเคลื่อนย้ายพร้อมกัน สมมุติว่าชื่อ Apple1, Apple2, Apple3 สามารถทำได้โดย

```
select -r Apple1 Apple2 Apple3;
move -x 5;
```

10. เราสามารถยกเลิกการ select วัตถุทุกชิ้นได้ด้วยคำสั่ง `select -cl`;

11. ในการเคลื่อนย้ายวัตถุ หมุนวัตถุ หรือ ปรับสัดส่วนวัตถุสามารถทำได้โดย

```
move -r 1 1 1 "ชื่อวัตถุ";
rotate -r 0 90 0 "ชื่อวัตถุ";
scale -r 0 1 0 "ชื่อวัตถุ";
```

โดย flag `-r` หมายถึงการคำนวณแบบ relative ถ้าไม่ต้องการให้เอา flag ออก

12. ถ้าต้องการ parent วัตถุสองชิ้นเข้าด้วยกันสามารถทำได้โดย

```
parentConstraint -mo "ชื่อวัตถุตัวแม่" "ชื่อวัตถุตัวลูก";
```

flag `-mo` หมายถึงการบังคับให้วัตถุทั้งสองตัวรักษาระยะห่างจากกัน (maintain offset)

Reference

- Wilkins, M. R. and Kazmier, C. (2005) *MEL Scripting for Maya Animations*, Morgan Kaufmann Publishers, Elsevier Inc.
- Galanakis, R. (2014) *Practical Maya Programming with Python*, Packt Publishing Ltd.
- Stripinis, D. (2003) *The MEL Companion: Maya Scripting 3D Artists*, Charles River Media, INC