Chapter 6:
Creating Maya Graphical User Interface (GUI)

Topics:

Part I: modal windows, confirm dialog, prompt dialog, progress window, window layouts, column layout, row layout, grid layout, pane layout, tab layout, frame layout, and scroll layout

Part II: window controls, field control types, field group controls, scroll and slider fields controls, colour slider group controls, scroll and slider fields controls, text scroll list controls, check box controls, radio button controls, and button controls

Chapter 6:
Creating Maya Graphical User Interface (GUI)

เอกสารประกอบการเรียน
รายวิชา ANI 951301
สาขาวิชาแอนนิเมชันและเกม
วิทยาลัยศิลปะ สื่อ และเทคโนโลยี
มหาวิทยาลัยเชียงใหม่

951301: Advanced Animation and Rendering Techniques

วัตถุประสงค์

1. เข้าใจถึงโครงสร้างการออกแบบและการทำงานของ graphical user interface (GUI) ภายในโปรแกรม Maya ด้วยการใช้ MEL scripting language
2. เข้าใจถึงหลักการทำงานและประโยชน์ของหน้าต่างเมนู window frame
3. เรียนรู้ถึง layouts แบบต่างๆที่สามารถสร้างได้ด้วย MEL scripting language และสามารถนำไปประยุกต์ใช้สร้างชุดเครื่องมือด้วยตนเองเบื้องต้นได้
4. เข้าใจถึงการออกแบบและสร้าง flow controls เพื่อใช้ในการควบคุมการทำงานของ scripts ได้
5. สามารถสร้างตัวควบคุม controls แบบต่างๆ และสามารถนำไปประยุกต์ใช้ด้วยตนเองได้อย่างเหมาะสม

เนื้อหาการสอน

ส่วนที่ 1: การออกแบบและสร้าง modal windows, confirm dialog, prompt dialog, progress window, window layouts, column layout, row layout, grid layout, pane layout, tab layout, frame layout, และ scroll layout

ส่วนที่ 2: การออกแบบและสร้าง window controls, field control types, field group controls, scroll and slider fields controls, colour slider group controls, scroll and slider fields controls, text scroll list controls, check box controls, radio button controls, และ button controls

Creating Maya Graphical User Interface (GUI)

รากฐานสำคัญของการสร้าง GUI คือการสร้างหน้าต่างไว้ใช้สำหรับโต้ตอบกับผู้ใช้ หน้าต่าง UI จะอยู่ในรูปของ floating panels ซึ่งสามารถเปิด ปิดได้ด้วยตัวผู้ใช้งานเองหรือด้วยการทำงานของ MEL command ในการใช้คำสั่ง MEL เพื่อสร้างหน้าต่างโต้ตอบกับผู้ใช้นั้นมีประโยชน์หลายประการ ในการทำงาน เช่นการสร้างชุดเครื่องมือใหม่ออกมา การสร้าง sliders เพื่อควบคุมการขยับกระดูกตัวละคร การสร้างแถบสีเพื่อเรียกใช้ รวบไปถึงการทำ flow control เพื่อใช้ในการตัดสินใจสร้างหรือขึ้นรูปวัตถุเป็นต้น ในบทเรียนนี้จะมุ่งเน้นให้นักศึกษาเข้าใจถึงแนวความคิดในการประยุกต์ใช้หน้าต่างในแบบต่างๆเพื่อเอื้อให้กับการทำงานต่อไป

ก่อนจะเริ่มต้นสร้างหน้าต่างนั้น เราควรต้องทราบโครงสร้างพื้นฐานในการออกแบบ UI ของ Maya ก่อน โดยโครงสร้างการออกแบบสามารถแบ่งออกได้เป็นสามส่วนหลักๆ ประกอบด้วย การสร้างโครงหน้าต่าง, การใส่เนื้อหาที่ต้องการลงไปในตัวหน้าต่าง และสุดท้ายคือการแสดงสิ่งที่สร้างไว้ออกมาในรูปหน้าต่างเครื่องมือ ในการทำงานด้วย MEL scripts จะต้องใช้โครงสร้างนี้ในการสั่งงาน เพื่อให้เห็นภาพ เราจะมาทดลองสร้างหน้าต่างอันแรกกันดังตัวอย่างด้านล่าง

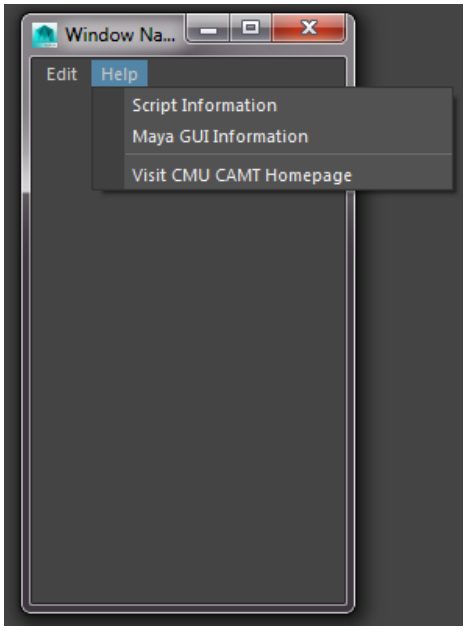

Fig 06-01: แสดงหน้าต่างแบบที่มี menu items

จากตัวอย่างด้านบนคือการสร้างหน้าต่างที่มีเมนูให้เลือกสองอันคือ edit และ help menus โดยที่แต่ละเมนูจะมี sub-menus อยู่ด้านใน ในตัวอย่างนี้ประกอบด้วย Script Information, Maya

951301: Advanced Animation and Rendering Techniques

GUI Information และ Visit CMU CAMT Homepage เป็น sub-menus ของ help โดยเราสามารถทำได้จากคำสั่งด้านล่างนี้

```
window // สั่งสร้างหน้าต่างเปล่าๆขึ้นมา
-title "Window Name" // ตั้งชื่อหน้าต่าง
-iconName "Icon Name" // ตั้งชื่อ icon ของ window
-widthHeight 200 350 // กำหนดขนาดของหน้าต่าง กว้าง x ยาว
-menuBar on // สร้าง menu bar เปล่า
-menuBarVisible on // กำหนดให้สามารถมองเห็น menu bar ได้
-minimizeButton on // อนุญาตผู้ใช้ให้กดปุ่ม minimize หน้าต่างได้
-maximizeButton on // อนุญาตผู้ใช้ให้กดปุ่ม maximize หน้าต่างได้
-sizeable on // อนุญาตผู้ใช้สามารถปรับขนาดของหน้าต่างได้
-titleBar on; // เปิดการแสดงผล title bar

// create first menu
menu // คำสั่งสร้างเมนู
-label "Edit" // ชื่อที่แสดงตรงเมนูว่า Edit
-tearOff false; // ไม่อนุญาตให้ลากเมนูออกมาเป็นหน้าต่างย่อยได้
menuItem // คำสั่งสร้าง item ให้กับเมนู
-label "Move"; // ตั้งชื่อ item ที่สร้างว่า move
menuItem
-label "Rotate";
menuItem
-label "Scale";

// create second menu
menu // สร้างเมนูอันที่สอง
-label "Help"
-tearOff on; // อนุญาตให้ลากเมนูออกมาเป็นหน้าต่างย่อยได้
menuItem
-label "Script Information";
menuItem
-label "Maya GUI Information";
menuItem
-divider 1; // สร้างตัวขั้นระหว่าง items
menuItem
-label "Visit CMU CAMT Homepage"
-command "showHelp -absolute\"http://www.camt.cmu.ac.th\"";
// สร้าง link ให้กับ item ตัวนี้ไปที่ website ของคณะ

showWindow; // ประกาศให้แสดงผลหน้าต่างที่สร้าง
```

จากคำสั่งด้านบนจะพบว่าเราต้องเริ่มต้นด้วยการใช้คำสั่ง window เพื่อสร้างหน้าต่างเปล่าๆขึ้นมาก่อน จากนั้นเราจะใส่รายละเอียดและองค์ประกอบของหน้าต่างเข้าไปเช่น ชื่อ ขนาด เมนู และ items เป็นต้น ถึงขั้นนี้ถ้าเรา execute คำสั่ง โปรแกรมจะสร้างหน้าต่างขึ้นมาแต่จะไม่แสดงผล คือเราจะไม่สามารถมองเห็นหรือโต้ตอบกับมันได้ จนกว่าเราจะประกาศให้โปรแกรมแสดงผลหน้าต่างที่สร้างขึ้นมาด้วยคำสั่ง showWindow ในขั้นตอนสุดท้าย

951301: Advanced Animation and Rendering Techniques

Modal Windows

ตอนนี้เราเข้าใจพื้นฐานการสร้างหน้าต่างเบื้องต้นแล้ว ขั้นต่อไปเราจะมาทำความรู้จักหน้าต่างแบบ modal windows กัน ซึ่งคือหน้าต่างที่รอการโต้ตอบจากผู้ใช้ก่อนที่มันจะกลับไปทำงานใน workflow ต่อไป ชื่ออื่นๆที่ถูกอ้างอิงถึงเช่น heavy windows และ modal dialog เนื่องจากบ่อยครั้งจะถูกใช้ในรูปแบบของ dialog box

หน้าต่าง modal windows สามารถแบ่งออกได้เป็นสามประเภทหลักๆคือ confirm dialog, prompt dialog และ progress window ทั้งสามประเภทจะมีหน้าที่การใช้งานและลักษณะของ input ที่แตกต่างกัน confirm dialog จะมีเพื่อให้ผู้ใช้คลิกยืนยันหรือยกเลิกการทำงานในขั้นต่อไป เช่นเมื่อ scripts ถูก run มาถึงจุดๆหนึ่งแล้วเราต้องการทราบว่าผู้ใช้ต้องการให้ทำงานต่อตาม workflow ในขั้นต่อไปหรือจะยกเลิกการทำงานเป็นต้น prompt dialog จะมีความแตกต่างกันตรงที่จะสามารถเก็บค่าที่ผู้ใช้ป้อนเข้าไปเพื่อนำไปใช้เป็นตัวแปรในการทำงานขั้นต่อไปได้ หรือสามารถป้อนข้อมูล input เพื่อใช้ในการทำงาน ประเภทสุดท้าย progress window จะเป็นหน้าต่างแสดงผลการทำงาน ณ ขณะนั้นว่าเป็นอย่างไร สำเร็จไปแล้วกี่เปอร์เซ็นต์ ต้องการจะให้ทำงานต่อไปหรือจะยกเลิกการทำงานเป็นต้น ทีนี้เราลองมาสร้าง modal windows ทั้งสามแบบกัน

1. Confirm Dialog

สามารถใช้เป็นเสมือนตัวป้องกันการผิดพลาดในการ run scripts โดยปกติแล้วผู้ออกแบบ scripts ส่วนมากมักจะใช้ confirm dialog เพื่อให้ผู้ใช้ยืนยันการทำงาน จัดอยู่ในพื้นฐานของการออกแบบ workflow โดยหน้าต่าง confirm dialog จะมีหน้าตาดังตัวอย่างด้านล่าง

Fig 06-02: แสดงหน้าต่างประเภท confirm dialog

เราสามารถสร้างหน้าต่าง confirm dialog ดังตัวอย่างได้ดังนี้

```
confirmDialog // สร้างหน้าต่าง confirm dialog
-title "Confirm Dialog" // ตั้งชื่อให้กับมัน
-message "What say you?" // ข้อความที่แสดงให้ผู้ใช้เห็น
-button "Yes" // สร้างปุ่มแรกพร้อมแสดงข้อความว่า Yes
-button "No" // สร้างปุ่มถัดมาพร้อมแสดงข้อความว่า No
-defaultButton "Yes" // กำหนดว่าถ้าผู้ใช้กดปุ่ม enter จะ return ค่าเป็น Yes
-cancelButton "No" // กำหนดว่าถ้าผู้ใช้กดปุ่ม escape จะ return ค่าเป็น No
-dismissString "No" // กำหนดว่าถ้าผู้ใช้กดปุ่มปิดหน้าต่าง (x) จะ return ค่าเป็น No
-backgroundColor 0.5 1 0.5; // กำหนดสีของหน้าต่างโดยให้ค่าเป็น RGB โดยมีค่าระหว่าง
0 - 1
```

2. Prompt Dialog

เมนูนี้จะถูกใช้เมื่อเราต้องการข้อมูล input เพียงอย่างเดียวจากผู้ใช้ โดยเฉพาะเมื่อ input นั้นอยู่ในรูปของตัวหนังสือ (texts) แทนที่จะประกาศเป็นค่าตัวแปรตั้งแต่แรก ผู้ใช้สามารถป้อนข้อมูลที่ต้องการลงไปได้ ซึ่งจะส่งผลให้ผู้ใช้เข้าใจได้ง่ายและเกิดความยืดหยุ่นเพิ่มขึ้นในการใช้งาน ซึ่งเป็นหัวใจในการออกแบบ scripts ซึ่งแตกต่างจาก confirm dialog เนื่องจาก prompt dialog ต้องการค่า input จากผู้ใช้ก่อนที่ data จะมีการเข้าถึงและถูกนำไปใช้ ในตัวอย่างด้านล่าง เราสร้าง prompt dialog เพื่อให้ผู้ใช้ป้อนข้อมูล website ที่จะให้โปรแกรม run ขึ้นมา

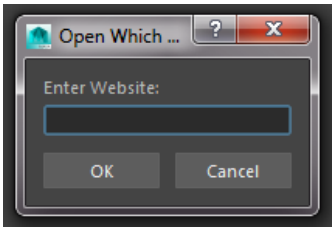

Fig 06-03: แสดงหน้าต่างประเภท prompt dialog

เราสามารถสร้างหน้าต่าง prompt dialog ดังตัวอย่างได้ดังนี้

```
string $URL; // ประกาศตัวแปรชื่อ $URL
string $result = `promptDialog // ประกาศตัวแปร $result ให้เก็บค่าจากหน้าต่าง
prompt dialog
-title "Which URL Do You Wanna Visit?" // ตั้งชื่อให้กับหน้าต่าง
-message "Enter Website:" // ข้อความที่จะปรากฏในหน้าต่าง
-button "OK" // สร้างปุ่ม OK ซึ่งมี string เป็นชื่อเดียวกัน
-button "Cancel" // สร้างปุ่ม Cancel ซึ่งมี string เป็นชื่อเดียวกัน
-defaultButton "OK" // กำหนดว่าถ้าผู้ใช้กดปุ่ม enter จะ return ค่าเป็น OK
-cancelButton "Cancel" // กำหนดว่าถ้าผู้ใช้กดปุ่ม escape จะ return ค่าเป็น
Cancel
```


951301: Advanced Animation and Rendering Techniques

```
-dismissString "Cancel"`; // กำหนดว่าถ้าผู้ใช้กดปุ่มปิดหน้าต่าง (x) จะ return ค่าเป็น Cancel

if ($result == "OK") // ตั้งเงื่อนไขว่าถ้าผลลัพธ์คือ OK จะให้...
{
    $URL = ("http://www.")+(`promptDialog -query`); // ... เปิดลิงค์ตามข้อความที่ผู้ใช้ป้อนเข้ามา (-query ดึงข้อมูลที่ป้อนมาใช้)
    showHelp -absolute $URL; //
}
```

3. Progress Window

เป็นการแสดงผลให้ผู้ใช้ทราบถึงการทำงานของ scripts แบบ real-time ว่าอยู่ในขั้นตอนไหน โดยจะแสดงผลในรูปแบบของ percentage bar พร้อมตัวเลข ในการเรียกใช้งาน progress window นั้น ผู้ออกแบบจะต้องทราบของเขตของ operations ทั้งหมดที่ต้องการวัดค่าก่อน ด้วยเหตุผลนี้จึงไม่เหมาะสมกับการทำงานร่วมกับ while loop แต่ควรจะใช้ for loop แทน ประโยชน์อีกอย่างของ progress window คือผู้ใช้สามารถสั่งยกเลิกการ run scripts ได้ตลอดเวลาถ้าเกิดใช้เวลานานเกินไป หรือต้องการแก้ไข scripts ที่มีความผิดพลาด ตัวอย่างของ progress window ดังตัวอย่างด้านล่าง

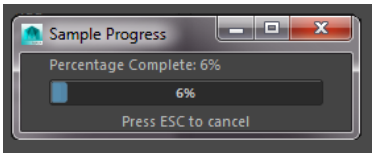

Fig 06-04: แสดงหน้าต่างประเภท progress window ที่แสดงผล progress bar

เราสามารถสร้างหน้าต่าง progress window ดังตัวอย่างได้ดังนี้ (เพื่อให้นักศึกษาเห็นถึงการทำงาน ตัวอย่างนี้ถึงสมมุติ for loop ขึ้นมาเพื่อให้โปรแกรมคำนวณ)

```
int $amount = 0; // ประกาศตัวแปรเตรียมไว้ใช้ใน for loop

progressWindow // สร้างหน้าต่าง progress window
-title "Sample Progress" // ตั้งชื่อให้กับหน้าต่าง
-progress $amount // กำหนดตัวแปรที่ต้องการให้วัดค่าคือ $amount
-status "Percentage Complete: 0%" // ข้อความ status ที่จะแสดงเหนือมาตรวัด
-isInterruptable true; // อนุญาตให้ผู้ใช้สามารถยกเลิกการ run ได้

for ( $amount = 0; $amount < 100; $amount++ ) // สร้าง for loop ขึ้นมาเพื่อเพิ่มค่า $amount
{
    if (`progressWindow -query -isCancelled`)
    break;
```


951301: Advanced Animation and Rendering Techniques

```
    progressWindow
    -edit
    -progress $amount
    -status ("Percentage Complete: "
    + $amount
    + "%"); // แสดงค่าตัวเลขจากค่า $amount
    pause -seconds 1;
}

progressWindow -endProgress; // terminate คำสั่งเมื่อค่าของ progress
value ถึงจุดสูงสุด
```

Layouts

การใช้คำสั่ง window เป็นการสร้างหน้าต่างเปล่าๆที่ยังไม่มีข้อมูลข้างในขึ้นมา ขั้นตอนต่อไปคือการออกแบบ layouts ให้กับหน้าต่างของเราว่าจะมีเนื้อหาอะไรบ้างและต้องการโต้ตอบกับผู้ใช้ในลักษณะไหน เนื้อหาในส่วนต่อไปนี้จะอธิบายถึงหน้าที่การทำงานและการออกแบบ layouts แบบต่างๆที่สามารถสร้างขึ้นใช้งานได้ ซึ่งประกอบด้วย column layout, row layout, grid layout, pane layout, tab layout, frame layout และ scroll layout ซึ่งแต่ละ layout จะมีคุณลักษณะแตกต่างกัน ขึ้นอยู่กับการใช้งานและข้อมูลที่ได้รับเป็นหลัก

1. Column Layout

จัดอยู่ใน layout แบบพื้นฐานที่สุด มีคุณลักษณะเฉพาะคือจะเรียงตัวเนื้อหาในเมนูตามแนวตั้ง จะถูกใช้เมื่อต้องการให้ผู้ใช้เลือกเพื่อ return ค่าการเลือกให้กับโปรแกรม หรือเพื่อ execute คำสั่งใดคำสั่งหนึ่งโดยตรงจาก options ที่มี จากตัวอย่างด้านล่างคือการสร้างเมนู เพื่อเรียกใช้คำสั่งในการ edit mesh แบบต่างๆซึ่งประกอบด้วยคำสั่ง add division, bevel, connect, detach, extrude และ merge to centre โดยในตัวอย่างนี้เราจะป้อนคำสั่งโดยตรงไว้ที่ปุ่มต่างๆในเมนู เมื่อผู้ใช้กดปุ่มใด คำสั่งนั้นจะถูกเรียกใช้ทันที นี่เป็นเพียงตัวอย่างหนึ่งของการใช้ column layout

951301: Advanced Animation and Rendering Techniques

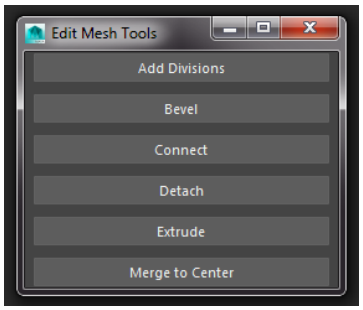

Fig 06-05: แสดงหน้าต่างประเภท column layout

เราสามารถสร้างหน้าต่าง column layout ออกแบบให้เป็น edit mesh tools ได้ดังตัวอย่างได้ดังนี้

```
window // สร้างหน้าต่าง
-title "Edit Mesh Tools"; // ตั้งชื่อว่า Edit Mesh Tools

columnLayout // สร้าง column layout
-columnAttach "both" 1 // กำหนดให้ขอบซ้ายและขวาของปุ่มล่นเข้ามาจากขอบซ้ายขวาเท่ากับ 1
-rowSpacing 1 // กำหนดช่องห่างระหว่างแต่ละปุ่มเป็น 1
-columnWidth 250; // กำหนดความกว้างให้กับ column

button // สร้างปุ่มแรก
-label "Add Divisions" // แสดงข้อความ Add Divisions ที่บนปุ่ม
-command polySubdivideFacet; // กำหนดคำสั่งที่จะ execute เมื่อผู้ใช้กดปุ่ม
button
-label "Bevel"
- command polyBevel;
button
-label "Connect"
-command polyConnectComponents;
button
-label "Detach"
-command DetachComponent;
button
-label "Extrude"
-command polyExtrudeFacet;
button
-label "Merge to Center"
-command MergeToCenter;

showWindow; // ประกาศให้แสดงผลหน้าต่างที่สร้าง
```

จากคำสั่งตัวอย่างจะพบว่าเราสามารถกำหนดคำสั่ง (-command) ที่จะถูกเรียกใช้ได้ทันทีเมื่อผู้ใช้กดปุ่มใดปุ่มหนึ่ง นอกจากนั้นเรายังสามารถสร้าง variables ขึ้นมาแทนแล้วใช้ loop ในการสั่งทำงานตามปุ่มที่เลือกเพื่อใช้ในการควบคุม workflow ของ scripts ก็สามารถทำได้

951301: Advanced Animation and Rendering Techniques

2. Row Layout

จัดอยู่ในพื้นฐานของ layout ทั้งหมดเช่นเดียวกับ column layout แตกต่างกันที่ row layout จะเรียงตัวปุ่มตามแนวนอน คุณลักษณะอื่นของ layout จะเป็นเช่นเดียวกัน ในตัวอย่างต่อไปเราจะทดลองออกแบบ row layout เพื่อใช้ในการขึ้นรูปทรงพื้นฐานประกอบด้วย cube, cylinder, sphere, prism และ torus กัน

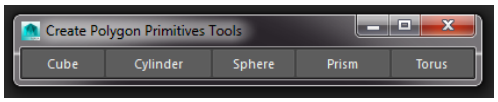

Fig 06-06: แสดงหน้าต่างประเภท row layout

เราสามารถสร้างหน้าต่าง row layout ออกแบบให้เป็น create polygon primitives tools ได้ดังตัวอย่างได้ดังนี้

```
window // สร้างหน้าต่าง
-title "Create Polygon Primitives Tools"; // ตั้งชื่อว่า Create Polygon Primitives Tools

rowLayout // สร้าง row layout
-numberOfColumns 5 // จำนวนปุ่มในหนึ่งบรรทัด (number of columns in the row)
-columnWidth5 70 90 70 80 70 // ความกว้างของแต่ละปุ่ม (ในที่นี้คือ 5 ปุ่ม) จากซ้ายมาขวา
-adjustableColumn 1 // กำหนดให้ปุ่มแรกสามารถปรับขนาดตามขนาดหน้าต่างได้
-columnAlign 1 "center" // กำหนดให้ตัวหนังสืออยู่ตรงกลาง
-columnAttach 1 "both" 0 // กำหนดให้ตัวปุ่มที่ 1 อ้างอิงขนาดกับทั้งซ้ายและขวา โดยให้มีช่องว่างระหว่างปุ่มเป็น 0
-columnAttach 2 "both" 0 // กำหนดให้ตัวปุ่มที่ 2 อ้างอิงขนาดกับทั้งซ้ายและขวา โดยให้มีช่องว่างระหว่างปุ่มเป็น 0
-columnAttach 3 "both" 0 // กำหนดให้ตัวปุ่มที่ 3 อ้างอิงขนาดกับทั้งซ้ายและขวา โดยให้มีช่องว่างระหว่างปุ่มเป็น 0
-columnAttach 4 "both" 0 // กำหนดให้ตัวปุ่มที่ 4 อ้างอิงขนาดกับทั้งซ้ายและขวา โดยให้มีช่องว่างระหว่างปุ่มเป็น 0
-columnAttach 5 "both" 0; // กำหนดให้ตัวปุ่มที่ 5 อ้างอิงขนาดกับทั้งซ้ายและขวา โดยให้มีช่องว่างระหว่างปุ่มเป็น 0

button // สร้างปุ่มแรก
-label "Cube" // แสดงข้อความ Cube บนปุ่ม
-command polyCube; // เมื่อกดปุ่มให้ run คำสั่ง polyCube

button // สร้างปุ่มที่สอง
-label "Cylinder" // แสดงข้อความ Cylinder บนปุ่ม
-command polyCylinder;

button // สร้างปุ่มที่สาม
```


951301: Advanced Animation and Rendering Techniques

```
-label "Sphere" // แสดงข้อความ Sphere บนปุ่ม
-command polySphere;

button // สร้างปุ่มที่สี่
-label "Prism" // แสดงข้อความ Prism บนปุ่ม
-command polyPrism;
button // สร้างปุ่มที่ห้า
-label "Torus" // แสดงข้อความ Torus บนปุ่ม
-command polyTorus;
showWindow; // ประกาศให้แสดงผลหน้าต่างที่สร้าง
```

3. Grid Layout

คุณลักษณะของ grid layout คือการรวม column กับ row layouts เข้าด้วยกัน ซึ่งมีประโยชน์เมื่อมีชุดคำสั่งมากๆ และไม่สามารถเรียงตามแนวแกนใดแกนหนึ่งได้พอ ตัวอย่างของ grid layout มีดังนี้

Fig 06-07: แสดงหน้าต่างประเภท grid layout

เราสามารถสร้าง grid layout ได้ด้วยคำสั่งด้านล่าง

```
window // สร้างหน้าต่าง
-title "gridLayout"; // ตั้งชื่อว่า gridlayout

gridLayout // สร้าง layout แบบ grid
-numberOfColumns 3 // กำหนดให้มีปุ่มตามแนวนอนได้สามอัน
-cellWidthHeight 90 50; // กำหนดขนาดของ layout
// ที่เหลือจะเหมือนกับการสร้างปุ่มใน layout แบบอื่นๆ ดังด้านล่าง
button
-label "First Button";
button
-label "Second Button";
button
-label "Third Button";
button
-label "Fourth Button";
button
-label "Fifth Button";
button
-label "Sixth Button";
showWindow; // ประกาศให้แสดงผลหน้าต่างที่สร้าง
```


951301: Advanced Animation and Rendering Techniques

4. Pane Layout

จากหัวข้อที่แล้วจะพบว่า grid layout นั้นตัวปุ่มจะมีขนาดคงที่ เมื่อเราปรับขนาดของหน้าต่าง ตัวปุ่มจะไม่ปรับขนาดตามแต่อย่างใด ซึ่งนี่เป็นข้อแตกต่างกับ pane layout ซึ่งตัวปุ่มจะปรับขนาดอัตโนมัติตามขนาดของหน้าต่าง ตัวอย่างด้านล่างแสดง pane layout ที่มีสี่ปุ่มปรับขนาดได้

Fig 06-08: แสดงหน้าต่างประเภท pane layout

เราสามารถสร้าง pane layout ได้ด้วยคำสั่งด้านล่าง เนื่องจากถึงจุดนี้นักศึกษาน่าจะมีความเข้าใจในตัวคำสั่งมากขึ้นแล้ว ทางผู้เขียนขอละเว้นคำอธิบายของชุดคำสั่งที่อธิบายไปแล้ว และจะอธิบายแต่คำสั่งใหม่ๆที่ยังไม่ได้กล่าวถึงแทน

```
window
-title "paneLayout"
-widthHeight 280 370;

paneLayout // สร้าง pane layout
-configuration "quad"; // คือการกำหนดการจัด layout ของ panes ซึ่งมีให้เลือกดังนี้ "single", "horizontal2", "vertical2", "horizontal3", "vertical3", "top3", "left3", "bottom3", "right3", "horizontal4", "vertical4", "top4", "left4", "bottom4", "right4", "quad" ในที่นี้จะใช้ quad ซึ่งคือหน้าต่างสี่อันเรียงกันแบบ grid

button
-label "First Button";
button
-label "Second Button";
button
-label "Third Button";
button
-label "Fourth Button";

showWindow;
```

5. Tab Layout

ประโยชน์ของ tab layout คือการที่เราสามารถจัดวาง controls จำนวนมากๆไว้ภายในหน้าต่างเดียวได้ โดยจะถูกจัดวางในลักษณะ tabs กั้นเป็นหน้าๆเพื่อแบ่งหมวดหมู่ของ controls ที่ต้องการ ในแต่ละ tab เราสามารถสร้าง column layout เพื่อจัดการกับตัวปุ่มด้านในได้ tab layout จะมีลักษณะดังภาพตัวอย่างด้านล่าง

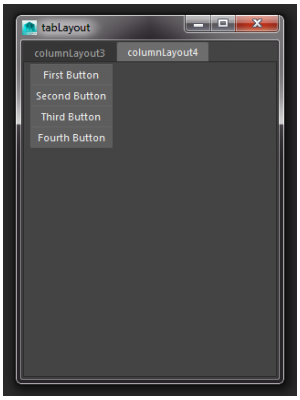

Fig 06-09: แสดงหน้าต่างประเภท tab layout

เราสามารถสร้าง tab layout ได้ด้วยคำสั่งด้านล่าง

```
window
-title "tabLayout"
-widthHeight 280 370;

tabLayout // สร้าง tab layout
-innerMarginWidth 5 // ขนาดความกว้างของ tab children
-innerMarginHeight 5; // ขนาดความสูงของ tab children

columnLayout // สร้าง layout ขึ้นมาใน tab แรก
-columnAttach "both" 5;
button
-label "First Button";
button
-label "Second Button";
button
-label "Third Button";
setParent..; // บอกกับโปรแกรมว่า layout ที่จะสร้างต่อไปจะไม่เป็นลูกของตัวนี้ ถ้าไม่กำกับ
layout ที่สร้างใหม่จะเป็น sub-layout ของตัวนี้

columnLayout // สร้าง layout ขึ้นมาใน tab ที่สอง
-columnAttach "both" 5;
button
-label "First Button";
button
-label "Second Button";
```


951301: Advanced Animation and Rendering Techniques

```
button
-label "Third Button";
button
-label "Fourth Button";
setParent..;

showWindow;
```

6. Frame Layout

เมื่อเรามี controls หลายตัวจนไม่สามารถจัดเก็บได้พอในหน้าเดียว หรือเราต้องการจัดเก็บคำสั่งให้เป็นหมวดหมู่ให้ชัดเจน เราสามารถจัดการได้ด้วย frame layout ข้อดีของ frame layout คือ เราสามารถกำหนดให้ซ่อนหรือแสดง children ในแต่ละหัวข้อได้ มีลักษณะคล้ายหน้าต่าง attribute editor ของ Maya นั่นเอง frame layout มีตัวอย่างดังนี้

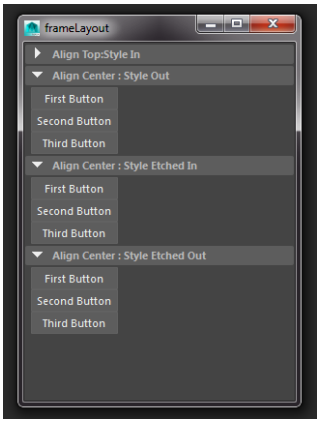

Fig 06-10: แสดงหน้าต่างประเภท frame layout

เราสามารถสร้าง frame layout ได้ด้วยคำสั่งด้านล่าง

```
window
-title "frameLayout"
-widthHeight 280 370;

columnLayout -adjustableColumn true; // สร้าง column layout แม่ให้สามารถปรับขนาดได้ เพื่อใช้เก็บ frames ทั้งหมด

frameLayout // สร้าง frame layout ขึ้นมาใน column แม่
-label "First Frame"
-labelAlign "top" // การวาง frame label สามารถเลือกได้เป็น top, center, bottom
-collapsable true;

// สร้าง column ย่อยขึ้นมาใน frame เพื่อเก็บปุ่ม
columnLayout
```


951301: Advanced Animation and Rendering Techniques

```
-columnAttach "both" 5;

// สร้างปุ่มขึ้นมาภายใน column ย่อย
button
-label "First Button";
button
-label "Second Button";
button
-label "Third Button";
setParent..; // เพื่อออกจาก column layout ย่อย
setParent..; // เพื่อออกจาก frame layout และกลับสู่ column layout ตัวแม่

frameLayout  // สร้าง frame layout ที่สองขึ้นมาใน column แม่
-label "Second Frame"
-labelAlign "center"
-borderStyle "out"
-collapsable true;

// สร้าง column ย่อยที่สองขึ้นมาใน frame layout ที่สอง เพื่อเก็บปุ่ม
columnLayout
-columnAttach "both" 5;

// สร้างปุ่มขึ้นมาใน column ย่อยที่สอง
button
-label "First Button";
button
-label "Second Button";
button
-label "Third Button";
setParent..; // เพื่อออกจาก column layout ย่อยที่สอง
setParent..; // เพื่อออกจาก frame layout ที่สอง และกลับสู่ column layout
ตัวแม่

// ขั้นตอนต่อไปจะเหมือนกับขั้นตอนที่ผ่านมา จึงของดคำอธิบายเพิ่มเติม
frameLayout
-label "Third Frame"
-labelAlign "center"
-borderStyle "etchedIn"
-collapsable true;

columnLayout
-columnAttach "both" 5;

button
-label "First Button";
button
-label "Second Button";
button
-label "Third Button";
setParent..;
```

```
setParent..;
frameLayout
-label "Fourth Frame"
-labelAlign "bottom"
-borderStyle "etchedOut"
-collapsable true;

columnLayout
-columnAttach "both" 5;


button
-label "First Button";
button
-label "Second Button";
button
-label "Third Button";
setParent..;
setParent..;

showWindow;
```

7. Scroll Layout

หัวข้อสุดท้ายคือ scroll layout ซึ่งมีลักษณะพิเศษคือจะมี scroll bar เพื่อใช้ในการ navigate ข้อมูลภายในหน้าต่าง มีลักษณะเหมือนกับ scroll bar ใน internet browser มีประโยชน์ในการจัดการกับข้อมูลที่มาก และไม่สามารถแบ่งเป็นหัวข้อย่อยๆได้ ตัวอย่างของ scroll layout มีดังนี้

Fig 06-11: แสดงหน้าต่างประเภท scroll layout

951301: Advanced Animation and Rendering Techniques

เราสามารถสร้าง scroll layout ได้ด้วยคำสั่งด้านล่าง

```
window
-title "scrollLayout"
-widthHeight 70 100;

scrollLayout // สร้าง scroll layout
-horizontalScrollBarThickness 10 // กำหนดขนาดของ scroll bar แนวนอน
-verticalScrollBarThickness 10; // กำหนดขนาดของ scroll bar แนวตั้ง

columnLayout
-columnAttach "both" 5;

// เพื่อเป็นการประหยัดเวลา เราจะใช้คำสั่ง for loop ในการสร้างปุ่มขึ้นมา 20 ปุ่ม
for ($i = 0; $i < 20; $i++)
button
-label ("Button " + ($i+1)); // ตั้งชื่อปุ่มตามค่า $i+1 เนื่องจากเราต้องการให้ปุ่มแรก
ชื่อ Button 1 ไม่ใช่ Button 0

showWindow;
```

Controls

ใน Maya นั้น ตัวควบคุมมีสองรูปแบบคือแบบพื้นฐาน กับแบบที่มีการรวมคุณลักษณะหลายอย่างไว้ในตัวควบคุมเดียว เพื่อให้การออกแบบ UIs มีความสะดวกขึ้น เนื้อหาในส่วนนี้จะครอบคลุมตัวควบคุมทั้งสองแบบ ตัวควบคุมบางตัวจะมีคุณลักษณะเฉพาะตัวที่สามารถทำงานกับค่าที่รับมาได้เฉพาะอย่าง ยกตัวอย่างเช่น text field control สามารถทำงานกับค่าตัวแปร (variables) แบบ string หรือ floating point ได้ แต่ float field control จะสามารถทำงานได้กับค่าตัวแปรแบบ floating point เท่านั้น การออกแบบต้องคำนึงถึงสิ่งที่เราต้องการจัดเก็บให้ดี และเลือกใช้ลักษณะของตัวควบคุมให้เหมาะสมกับสิ่งที่ต้องการ

Field Control Types

ตัวควบคุมสามารถแบ่งได้เป็นสามประเภทหลักๆตามชนิดของ variables ที่สามารถทำงานด้วย ซึ่งประกอบด้วย text field control, integer field control และ floating point field control ในการออกแบบหนึ่งหน้าต่างเมนู เราสามารถใช้ตัวควบคุมมากกว่าหนึ่งประเภทรวมกันได้ เราลองมาทำความรู้จักตัวควบคุมทั้งสามประเภทโดยการสร้างหน้าต่างเมนูที่มีการใช้ตัวควบคุมหลายๆแบบกันดังตัวอย่างด้านล่าง

951301: Advanced Animation and Rendering Techniques

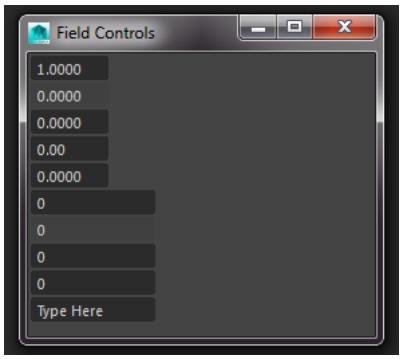

Fig 06-12: แสดงหน้าต่างที่ประกอบด้วย text field control, integer field control, และ floating point field control

จากตัวอย่างด้านบนจะสังเกตได้ว่าประกอบด้วยตัวควบคุมทั้งแบบ float field, integer field และ text field รวมถึงตัวควบคุมแบบที่แก้ไขได้ (ช่องสีเทาเข้ม) และแบบที่ไม่สามารถแก้ไขได้ (ช่องสีเทาอ่อน) คำสั่งในการสร้างหน้าต่างตัวควบคุมมีดังนี้

```
window // สร้างหน้าต่าง
-title "Field Controls" // ตั้งชื่อ
-widthHeight 190 500; // กำหนดขนาด

columnLayout; // สร้าง column layout มาจัดเก็บตัวควบคุมตามแนวตั้ง

// สร้าง float field อันแรก ให้มีคุณลักษณะเหมือนค่าเริ่มต้น (default)
floatField;

// สร้าง float field อันที่สอง ให้มีคุณลักษณะเหมือนค่าเริ่มต้น (default) แต่ไม่อนุญาตให้มีการแก้ไขค่า
floatField
-editable false;

// สร้าง float field อันที่สาม โดยกำหนดให้ผู้ใช้สามารถใส่ค่าลงไปได้อยู่ระหว่าง -20 ถึง 20 (ถ้าน้อยกว่าหรือมากกว่า โปรแกรมจะไม่รับ)
floatField
-minValue -20 // ค่าน้อยสุดที่อนุญาต
-maxValue 20 // ค่ามากสุดที่อนุญาติ
-value 0; // กำหนดค่าเริ่มต้น

floatField
-minValue 0
-maxValue 1
-precision 2; // จำนวนทศนิยมให้มีสองตำแหน่ง

floatField
-minValue -1
-maxValue 1
-precision 4
-step 0.01; // จำนวนค่าที่จะเปลี่ยนแปลง เมื่อมีการเลื่อน slider bar control
// สร้าง integer field ให้มีคุณลักษณะเหมือนค่าเริ่มต้น (default)
```


951301: Advanced Animation and Rendering Techniques

```
intField;

// สร้าง integer field ให้มีคุณลักษณะเหมือนค่าเริ่มต้น (default) แต่ไม่อนุญาตให้มีการแก้ไขค่า
intField
-editable false;

intField
-minValue -10
-maxValue 10
-value 0;

intField
-minValue -1000
-maxValue 1000
-step 10;

// สร้าง text field โดยให้แสดง Type Here เป็นข้อความที่ปรากฏตอนแรก
textField
-text "Type Here";
showWindow;
```

1. Field Group Controls

ในการใช้งาน field group controls เราสามารถสร้าง fields ทีละหลายๆ fields ด้วยคำสั่งเดียว ประโยชน์อีกประการคือการจัดการกับ data เฉพาะบางอย่างเช่นค่า translation ของวัตถุที่จะมีการอ้างอิงค่าสามค่าตามแนวแกน X, Y และ Z เป็นต้น ตัวอย่างของ field group controls มีดังนี้

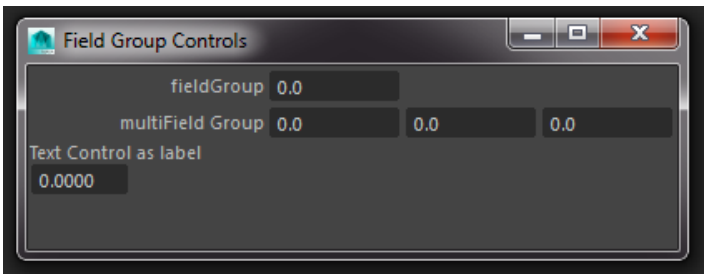

Fig 06-13: แสดงตัวควบคุมประเภท field group controls

จากตัวอย่างด้านบนเราสร้าง field group ขึ้นมาสามแบบ คือแบบ field group, multiple field group และ text control ซึ่งแต่ละแบบจะมีประโยชน์ในการใช้งานแตกต่างกัน เราสามารถสร้าง field group controls ได้ด้วยคำสั่งดังต่อไปนี้

951301: Advanced Animation and Rendering Techniques

```
window
-title "Field Group Controls"
-widthHeight 390 110;

columnLayout;

floatFieldGrp // สร้าง field group อันแรกแบบ floating point
-numberOfFields 1 // กำหนดให้มีจำนวน field = 1
-label "fieldGroup"; // ข้อความที่แสดงตรง field

floatFieldGrp // สร้าง field group อันที่สองแบบ floating point
-numberOfFields 3 // กำหนดให้มีจำนวน fields = 3
-label "multiField Group"; // ข้อความที่แสดงตรง field

text // สร้าง text field
-label "Text Control as label"; // ข้อความที่แสดงตรง field
textField; // กำหนดให้ประเภทของข้อความที่เติมได้เป็นแบบ text

showWindow;
```

2. Scroll and Slider Fields Controls

Scroll field และ slider field มีลักษณะคล้ายกันอย่างมาก แตกต่างกันที่ slider field จะสร้าง slider bar ที่เราสามารถปรับเปลี่ยนค่าได้ด้วยการเลื่อนแถบควบคุมไปทางซ้ายและขวา ในขณะที่ scroll field จะมีส่วนของหัวลูกศรที่ปลายทั้งสองด้านของ slider bar คล้ายกับในหน้าต่าง web browser เพื่อให้ผู้ใช้สามารถเลื่อนแถบควบคุมแบบ large step ได้ ลักษณะของ scroll field และ slider field ดังตัวอย่างด้านล่าง

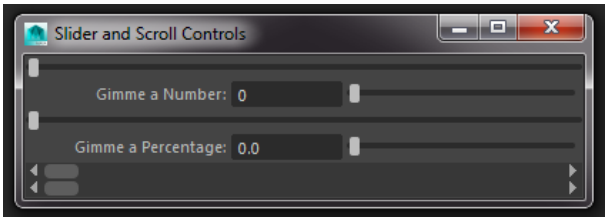

Fig 06-14: แสดงตัวควบคุมประเภท scroll และ slider fields controls

จากตัวอย่างด้านบน เราสร้าง integer slider, integer slider group (มีตัวเลขบอกค่า), float slider, float slider group (มีตัวเลขทศนิยมบอกค่า) และ float scroll bar สองอัน โดยถูกสร้างมาจากคำสั่งดังนี้

951301: Advanced Animation and Rendering Techniques

```
window
-title "Slider and Scroll Controls";

columnLayout
-adjustableColumn true;

// first field
intSlider; // สร้าง slider field เพื่อรับค่าแบบ integer

// second field
intSliderGrp // สร้าง slider group เพื่อรับค่าแบบ integer
-min 0 // กำหนดค่าต่ำสุดเป็น 0
-max 100 // กำหนดค่าสูงสุดเป็น 100
-value 0 // กำหนดค่าเริ่มต้นเป็น 0
-step 1 // เมื่อเลื่อน slider จะเปลี่ยนค่าเป็น 1
-label "Gimme a Number:" // ข้อความที่แสดงบน field
-field true; // อนุญาตให้ป้อนค่าด้วยตัวเลขได้

// third field
floatSlider; // สร้าง slider field เพื่อรับค่าแบบ floating point

// fourth field
floatSliderGrp // สร้าง slider group เพื่อรับค่าแบบ floating point
-min 0
-max 100
-value 0
-step 1
-label "Gimme a Percentage:"
-field true;

// fifth field
floatScrollBar; // สร้าง scroll field เพื่อรับค่าแบบ floating point

// sixth field
floatScrollBar // สร้าง scroll field เพื่อรับค่าแบบ floating point
-min 0
-max 100
-value 0
-step 1
-largeStep 10; // กำหนดค่าการเปลี่ยนแปลงแบบ large step เมื่อกดไปที่หัวลูกศรที่ตำแหน่ง
หัวท้ายของ bar

showWindow;
```


951301: Advanced Animation and Rendering Techniques

3. Colour Slider Group Controls

เราสามารถใช้ colour slider group เพื่อควบคุมสีที่เราต้องการได้ สามารถทำงานได้ทั้งในระบบ RGB และ HSV ขึ้นอยู่กับการออกแบบของผู้สร้าง จากตัวอย่างด้านล่างเราสร้างแถบควบคุมสีด้วยระบบ RGB ขึ้นมาสี่แถบ โดยที่แถบบนสุดจะควบคุมค่า hue, แถบที่สองควบคุมค่า R, แถบที่สามควบคุมค่า G, และแถบสุดท้ายควบคุมค่า B ตามลำดับ

Fig 06-15: แสดงตัวควบคุมประเภท colour slider group controls

เราสามารถสร้าง colour slider control ได้ด้วยคำสั่งด้านล่าง

```
window
-title "colorSlider Controls";
columnLayout
-adjustableColumn true;

colorIndexSliderGrp // สร้าง slider ควบคุมสีที่ 1
-label "Hue" // ตั้งชื่อว่า Hue
-min 0 // ค่าเริ่มต้นที่ 0
-max 20 // ค่าสูงสุดที่ 20
-value 10; // ค่า colour index (กำหนดว่าจะเริ่มต้นให้เข็มชี้ไปที่สีอะไร)

colorSliderButtonGrp // สร้าง slider ควบคุมสีที่ 2
-label "Colour R" // ตั้งชื่อว่า Colour R
-buttonLabel "Button" // แสดงข้อความว่า Button ที่ปุ่มควบคุม
-rgb 1 0 0; // ใช้ระบบ RGB โดยให้ R=1, G=0, B=0

colorSliderButtonGrp // สร้าง slider ควบคุมสีที่ 3
-label "Colour G" // ตั้งชื่อว่า Colour G
-buttonLabel "Button" // แสดงข้อความว่า Button ที่ปุ่มควบคุม
-rgb 0 1 0; // ใช้ระบบ RGB โดยให้ R=0, G=1, B=0

colorSliderButtonGrp // สร้าง slider ควบคุมสีที่ 4
-label "Colour B" // ตั้งชื่อว่า Colour B
-buttonLabel "Button" // แสดงข้อความว่า Button ที่ปุ่มควบคุม
-rgb 0 0 1; // ใช้ระบบ RGB โดยให้ R=0, G=0, B=1

showWindow; // แสดงหน้าต่างที่สร้าง
```

4. Text Scroll List Controls

มีไว้จัดการกับข้อมูลแบบ text เพื่อใช้ในการ list ข้อความแบบต่างๆ เช่นชื่อของวัตถุ ชื่อของแหล่งกำเนิดแสง หรือ ชื่อของข้อต่อเป็นต้น เราสามารถออกแบบใช้ผู้ใช้เลือกคลิกได้ทีละหลายๆ ชื่อ หรือบังคับให้เลือกได้ทีละชื่อก็ได้ เหมาะที่จะใช้ในสถานการณ์ที่ต้องจัดการกับ string array ทีละมากๆ ยกตัวอย่างเช่นเมื่อต้องการเข้าถึงข้อต่อทุกๆอันในหน้าต่างการทำงาน เป็นต้น

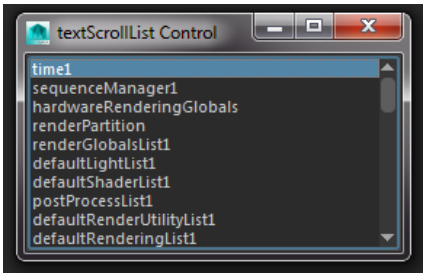

Fig 06-16: แสดงตัวควบคุมประเภท text scroll list controls

จากตัวอย่าง เราจะสร้าง text scroll list ไว้สำหรับแสดงรายชื่อของๆทุกอย่างที่อยู่ภายใน scene เพื่อดูว่าประกอบด้วยอะไรบ้าง เราสามารถทำได้ด้วยคำสั่งด้านล่าง

```
string $sceneList[] = `ls`; // ประกาศตัวแปรเพื่อเก็บค่าองค์ประกอบทุกอย่างภายใน
scene

window
-title "textScrollList Control";

columnLayout
-adjustableColumn true;

string $scroll = `textScrollList // สร้าง text scroll list
-numberOfRows 10 // จำนวนบรรทัดที่มองเห็นได้
-allowMultiSelection true`; // อนุญาตให้มีการเลือกของใน list ได้มากกว่าหนึ่งอัน

// นำค่าที่ได้จาก $sceneList มาแสดงใน text scroll list ที่สร้าง
for ( $each in $sceneList )
textScrollList
-edit
-append $each
$scroll;

showWindow;
```


951301: Advanced Animation and Rendering Techniques

5. Check Box Controls

Check box มีไว้เพื่อควบคุมค่า Boolean attribute หรือ variables ต่างๆ การสร้าง check box สามารถสร้างให้อยู่ในรูปแบบของ group ที่มีสมาชิคไม่เกินสี่ได้ ตัวอย่างถัดไปแสดงหน้าตาของ check box window ที่แสดงชื่อของ camera ภายใน scene

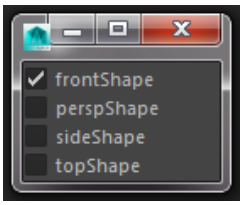

Fig 06-17: แสดงตัวควบคุมประเภท check box controls

เราสามารถสร้าง check box ได้ด้วยคำสั่งด้านล่าง

```
string $camList[] = `ls -type "camera"`; // ประกาศตัวแปรเพื่อเก็บค่าของ camera ภายใน scene ให้อยู่ใน $camList[]

window
-title "Check Box Controls";

columnLayout
-adjustableColumn true;

for ($each in $camList)
checkBox -label $each; // สร้าง check box โดยให้แสดงข้อความตามชื่อของ camera ใน $camList[]

showWindow;
```

6. Radio Button Controls

จะถูกแสดงในรูปของช่องให้เลือกใช้แบบกลมๆ จะมีลักษณะคล้ายกับ check box ที่เป็นช่องสี่เหลี่ยมแต่มีความพิเศษคือ ผู้ใช้สามารถเลือกได้ทีละหนึ่งตัวเลือกในกลุ่มเท่านั้น ถ้าเราเลือกมากกว่าหนึ่งตัว โปรแกรมจะยกเลิกการเลือกอันก่อนหน้า แล้วเลือกที่อันสุดท้ายเท่านั้น เหมาะกับการทำงานร่วมกับ switch case ในส่วนของ conditional statement มีหน้าตาดังตัวอย่างด้านล่าง

951301: Advanced Animation and Rendering Techniques

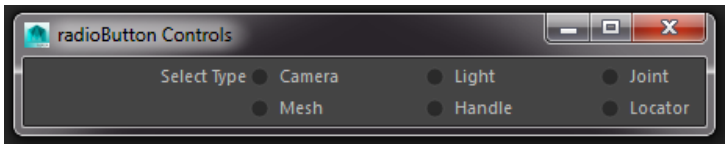

Fig 06-18: แสดงตัวควบคุมประเภท radio button controls

เราสามารถสร้าง radio buttons ได้ด้วยคำสั่งด้านล่าง

```
window
-title "radioButton Controls";

columnLayout
-adjustableColumn true;

string $group1 = `radioButtonGrp // สร้าง radio buttons บรรทัดแรก
-numberOfRadioButtons 3 // กำหนดให้มีสมาชิกสามตัว
-label "Select Type" // แสดงข้อความว่า select type ที่ด้านหน้า
-labelArray3 "Camera" "Light" "Joint"`; // สร้าง string สามตัวชื่อว่า
Camera, Light และ Joint พร้อมแสดงข้อความ

radioButtonGrp // สร้าง radio buttons บรรทัดที่สอง
-numberOfRadioButtons 3 // กำหนดให้มีสมาชิกสามตัวเช่นกัน
-shareCollection $group1 // ประกาศว่า radio buttons กลุ่มนี้จะมีความเกี่ยวข้อง
กับบรรทัดแรก
-label "" // ไม่แสดงผลข้อความใดๆด้านหน้า
-labelArray3 "Mesh" "Handle" "Locator"; // สร้าง string สามตัวชื่อว่า
Mesh, Handle และ Locator พร้อมแสดงข้อความ

showWindow;
```

7. Button Controls

เป็นตัวควบคุมที่ถูกใช้บ่อยมากในการออกแบบ button controls สามารถแบ่งได้เป็นสองประเภทหลักๆคือแบบ text label และแบบ symbol button แบบ text label จะแสดงข้อความตัวหนังสือลงบนปุ่ม ส่วนแบบ symbol button จะแสดงข้อความภาพได้ เราสามารถใช้ symbol button สร้างชุดแถบเครื่องมือของเราขึ้นมาเองได้ ซึ่งจะมีข้อได้เปรียบคือเราสามารถใช้ icon ที่มีขนาดใดก็ได้ ไม่ได้ถูกจำกัดที่ 32x32 pixel ดังเช่น icon บน shelf ขอแนะนำให้เราจัดเก็บไฟล์ภาพที่ใช้ไว้ในที่เดียวกันอย่างเป็นระบบ เพื่อป้องกันความสับสนในอนาคต ตัวอย่างของ button controls ทั้งสองแบบดังตัวอย่างถัดไป

951301: Advanced Animation and Rendering Techniques

Fig 06-19: แสดงตัวควบคุมประเภท button controls

เราสามารถสร้าง button controls ได้ด้วยคำสั่งด้านล่าง

```
window
-title "button Controls";

rowLayout
-numberOfColumns 15; // จำนวนปุ่มในหนึ่งบรรทัด (number of columns in
the row)

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/column.png";
// บอกให้โหลดไฟล์ภาพจาก root ของเครื่อง

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/row.png";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/grid.png";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/pane.png";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/tab.png";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/frame.png";
// บอกให้โหลดไฟล์ภาพจาก root ของเครื่อง

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/scroll.png";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/field.png";

symbolButton // สร้างปุ่มแบบ symbol button
```


951301: Advanced Animation and Rendering Techniques

```
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/fieldgrp.png
";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/scrollslider
.png";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/colour.png";
// บอกให้โหลดไฟล์ภาพจาก root ของเครื่อง
symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/textscroll.p
ng";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/checkbox.png
";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/radio.png";

symbolButton // สร้างปุ่มแบบ symbol button
-image
"C:/Users/CAMT/Documents/maya/2016/prefs/icons/button.png";

showWindow;
```

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

951301: Advanced Animation and Rendering Techniques