Chapter 4:
Grass Generator

Topics:

Part I: grass properties, shape deformation and bending attributes, and blades creation

Part II: grass random creation, generating the grasses, generation density

Chapter 4:
Grass Generator

เอกสารประกอบการเรียน
รายวิชา ANI 951301
สาขาวิชาแอนนิเมชันและเกม
วิทยาลัยศิลปะ สื่อ และ
เทคโนโลยี
มหาวิทยาลัยเชียงใหม่

951301: Advanced Animation and Rendering Techniques

วัตถุประสงค์

1. สามารถประยุกต์ใช้รูปทรงพื้นฐาน (polygon primitives) ในการสร้างสรรค์งานที่ต้องการได้ เช่นการดัดแปลงรูปทรงกรวยในการสร้างต้นหญ้า
2. ใช้การ deform และ bend shape ในการ random รูปทรงให้กับต้นหญ้าได้
3. สามารถสร้างการ generate ต้นหญ้าแบบ random ตำแหน่งและทิศทางของลำต้น
4. สามารถควบคุมความหนาแน่น (density) และพื้นที่ (area) ให้กับการสร้างสนามหญ้า polygon ได้

เนื้อหาการสอน

ส่วนที่ 1: คุณลักษณะทางกายภาพของต้นหญ้า, การปรับเปลี่ยนรูปทรง, การปรับแต่งค่าการดัดงอให้กับรูปทรง, และ การสร้างรูปทรงต้นหญ้า

ส่วนที่ 2: การกำหนดตำแหน่งการเกิดต้นหญ้าแบบสุ่ม, การสร้างทุ่งหญ้า, และการกำหนดความหนาแน่นให้กับทุ่งหญ้า

Grass Generator

เนื้อหาส่วนต่อไปเราจะมาทดลองนำความรู้ที่ได้มาประยุกต์ใช้ในทางที่มีความท้าทายมากขึ้น นั่นคือการสร้างสิ่งที่เป็นธรรมชาติอย่างทุ่งหญ้ากัน การสร้างสิ่งเลียนแบบธรรมชาติในเชิง computer graphic ถือเป็นสิ่งที่ท้าทาย เนื่องจากธรรมชาติมีความซับซ้อนและมีรายละเอียดสูง อย่างเช่นการทำทุ่งหญ้า ต้นหญ้าที่มีขนาดแตกต่างกัน หันไปในทิศทางไม่เหมือนกัน เกิดขึ้นอย่างไม่สม่ำเสมอบนพื้นเดียวกัน เป็นสิ่งที่เราต้องนำมาคิด การทำด้วยวิธีแบบปกติด้วย Maya ต้องใช้เวลาและความอดทนอย่างมาก เพราะต้องสร้างความเป็นเอกลักษณ์ให้กับต้นหญ้าในแต่ละต้นเลยทีเดียว ซึ่งในที่นี้การใช้ MEL เข้ามาช่วย จะสามารถลดเวลาในการทำงานลงได้มาก

Fig 04-01: ภาพตัวอย่างทุ่งหญ้าที่สร้างจากการใช้วิธีการในบทเรียน

ตัวอย่างด้านบนคือทุ่งหญ้าแบบที่เรากำลังจะทำกันในวันนี้ ในการนั้นเราต้องสร้างความแตกต่างให้กับต้นหญ้าแต่ละต้นในเรื่องของ ความยาว, ความโค้ง และ ความหนาแน่นต่อหนึ่งหน่วยพื้นที่ ก่อนที่เราจะมาเริ่มทำทุ่งหญ้าได้นั้น เรามาเรียนรู้การสร้างต้นหญ้าแต่ละต้นกันก่อน

Grass Properties

การหาข้อมูลของต้นหญ้านั้น เราสามารถทำได้ง่ายๆด้วยการออกไปดูต้นหญ้าแบบต่างๆที่ขึ้นอยู่รอบๆตัวเรา ความน่าสนใจของธรรมชาติอยู่ที่ความแตกต่างของรายละเอียดที่มันมี กับรายละเอียดที่ตาเราสามารถมองเห็นได้ เมื่อมองจากที่ไกล เราจะเห็นต้นหญ้ารวมเป็นพื้นที่สีเขียวเดียวกัน แต่ถ้าเราลองเข้าไปดูอย่างใกล้ชิดทีละต้น เราจะพบสีของลำต้นที่มีความแตกต่างกัน ลักษณะของใบหญ้าที่เมื่อดู

951301: Advanced Animation and Rendering Techniques

ใกล้ๆจะพบว่ามีลักษณะไม่ต่างอะไรกับใบมีด ความโค้งของใบหญ้าเกิดจากการที่ลำตัวของมันไม่สามารถรับน้ำหนักในส่วนยอดของใบได้ จึงเกิดเป็นความโค้งที่ลดหลั่นกันลงมา ลักษณะเฉพาะตัวเหล่านี้ คือสิ่งที่เราต้องสังเกตและจดจำเพื่อมาประยุกต์ใช้ในการทำงานศิลปะ เมื่อเราได้ข้อมูลมาเพียงพอแล้ว เราจะมาเริ่มทดลองสร้างต้นหญ้าใน workshop 2 กัน

Workshop 1

ในการทำงานนั้น เราไม่มีความจำเป็นที่ต้องเริ่มทุกอย่างจากศูนย์ แต่เราควรมองถึงสิ่งต่างๆที่ทาง Maya มีให้ไว้เป็นอุปกรณ์พื้นฐาน แล้วประยุกต์ใช้ให้เหมาะสมกับงานที่เราจะทำ ดังที่กล่าวไว้แล้วข้างต้นว่าใบหญ้านั้นมีลักษณะคล้ายใบมีด ทีนี้เราลองมาดูในรูปทรงพื้นฐาน (polygon primitives) ว่ามีอะไรที่มีขนาดใกล้เคียงที่สุดแล้วเราจะดัดแปลงมันอย่างไรให้ได้สิ่งที่เราต้องการ

ถ้าลองไล่ดูรูปทรงทั้งหมดเราจะพบว่ารูปทรงกรวย (cone) นั้นมีความใกล้เคียงอยู่มาก มันมีฐานกว้างและมียอดแหลม ถ้าเราสามารถเพิ่มขนาดความยาวให้กับมัน และบีบมันให้มีลักษณะแบนเราจะได้รูปทรงคล้ายใบมีดได้ไม่ยาก ข้อดีในการสร้าง cone คือเราสามารถกำหนด subdivisions ในแนวแกน X ให้กับมันได้ ถ้าเราลดมันลงเหลือ 3 เราจะได้รูปทรงคล้ายกับรูปพีรามิดนั่นเอง

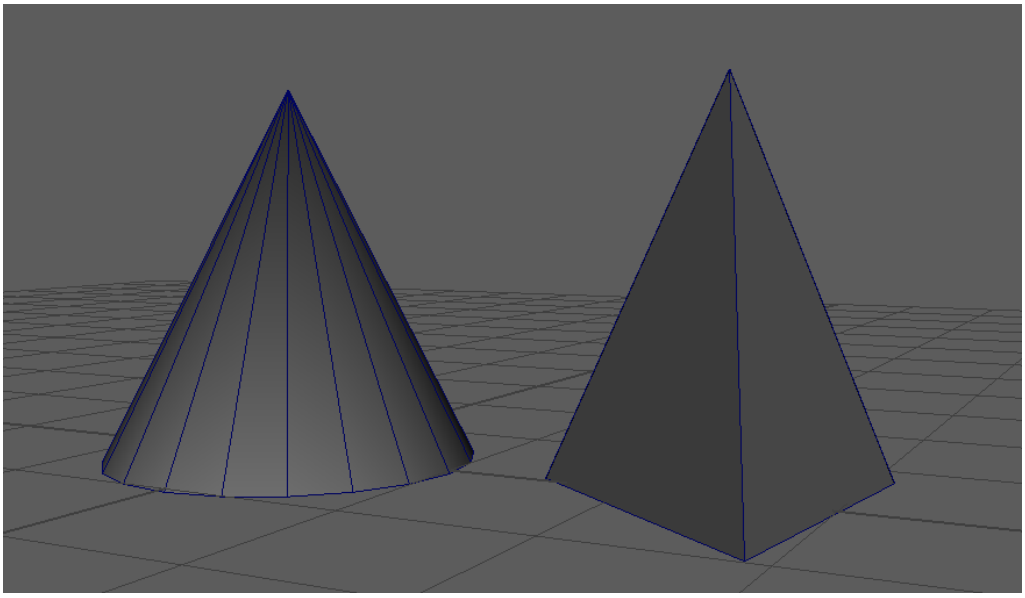

Fig 04-02: การตั้งค่า subdivisions X ของรูปทรงกรวยให้เหมาะสมกับการสร้างต้นหญ้า

951301: Advanced Animation and Rendering Techniques

จากภาพตัวอย่างแสดงการสร้าง polygon cones ในแบบ default ที่ subdivisions X = 20 ทางด้านซ้าย เปรียบเทียบกับรูปทรงที่ปรับค่า subdivisions X = 3 ทางด้านขวา

ทีนี้เราลองมาปรับแต่ง cone ด้านขวาให้มีลักษณะคล้ายใบมีดของต้นหญ้ากัน เริ่มจากการปรับให้มันมีลักษณะแบนก่อน สามารถทำได้โดยการปรับค่า scale ตามแนวแกน X ของมันจาก 1 เป็น 0.25 จากนั้นเพิ่มความยาวให้กับมันโดยการเพิ่มค่า scale ตามแนวแกน Y ให้เหมาะสมตามต้องการ ในที่นี้จะใช้ค่า scale Y = 10 เราจะได้ cone ที่มีลักษณะคล้ายใบมีดต้นหญ้าดังภาพตัวอย่างถัดไป

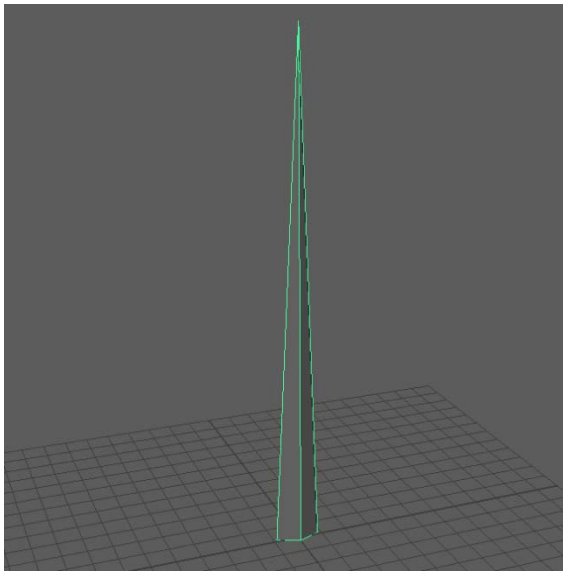

Fig 04-03: การปรับแต่งรูปทรงกรวยให้มีลักษณะใกล้เคียงกับต้นหญ้า

เมื่อได้ใบมีดมาเป็นที่น่าพอใจแล้ว ขั้นตอนต่อไปเราจะทำการดัดมันให้เกิดความโค้งเช่นเดียวกับใบหญ้า ก่อนอื่นเราต้องเตรียมความพร้อมให้กับวัตถุของเราให้เอื้อต่อการดัดโค้ง โดยการเพิ่มค่า subdivisions ตามแนวแกน Y จากเดิม 1 เป็น 20 ก่อน จากนั้นให้เปิด menu มากดัดวัตถุโดยไปที่โหมด Animation จากนั้นไปที่ Anim Deform>Open Full Deform Menu เพื่อเปิดหน้าต่างเครื่องมือ deform ขึ้น จากหน้าต่าง deform ให้เลือกไปที่ Nonlinear>Bend [] ให้คลิกไปที่เครื่องหมายสี่เหลี่ยมด้านหลังคำว่า Bend เพื่อเปิดหน้าต่างการตั้งค่าการดัดขึ้น ซึ่งหน้าต่างเครื่องมือ deform กับหน้าต่าง Bend มีลักษณะดังภาพตัวอย่างด้านล่าง

951301: Advanced Animation and Rendering Techniques

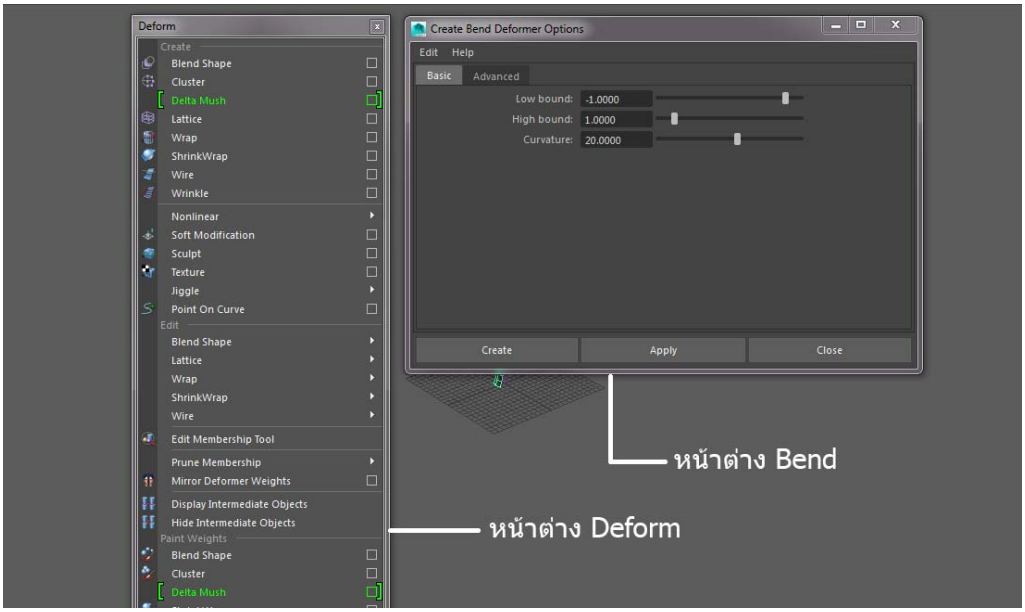

Fig 04-04: การเรียกใช้คำสั่ง bend deformer

จากนั้นให้ตรวจสอบให้แน่ใจว่าวัตถุที่ต้องการดัดของเราถูก selected อยู่ ให้ปรับค่า Curvature ในหน้าต่าง Bend เป็น 20 แล้วทดลองกด Apply ดู จะพบว่าวัตถุของเราถูกดัดให้มีความโค้งดังภาพตัวอย่างด้านล่าง

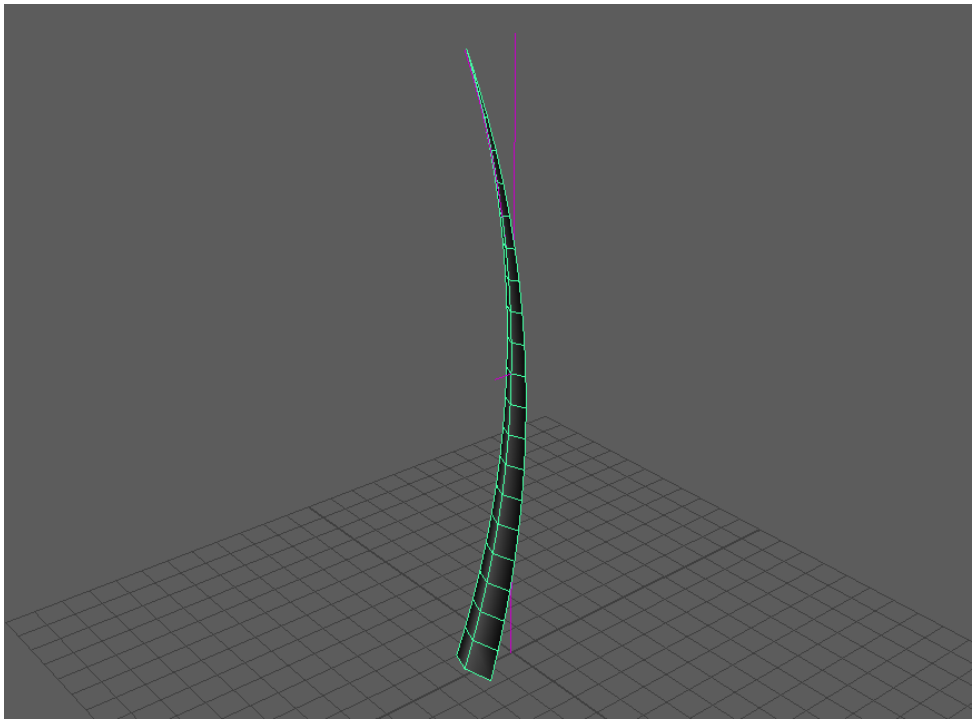

Fig 04-05: ต้นหญ้าที่ถูกดัดให้มีความโค้ง

951301: Advanced Animation and Rendering Techniques

จากขั้นตอนทั้งหมดนักศึกษาคงจะสามารถจินตนาการขั้นตอนต่างๆที่เราจะเขียนคำสั่งใน MEL ได้เบื้องต้น ต้องขอย้ำว่านี่เป็นเพียงขั้นตอนบางส่วนเท่านั้น ซึ่งยังไม่รวมถึงการปรับตำแหน่ง pivot ให้อยู่ตรงโคนต้น การปรับองศาของลำต้นให้ตั้งฉากกับพื้นดิน และการปรับทิศทางและความโค้งของต้นหญ้าในแต่ละต้นให้มีความแตกต่างกัน รวมถึงการสร้างและจัดวางต้นหญ้าแบบนี้ขึ้นมา 1000 – 4000 ต้นต่อ 10 ตาราง units ของ Maya นี่คือเหตุผลที่นักสร้าง animation มืออาชีพส่วนมากจะใช้ MEL เข้ามาเป็นส่วนร่วมในการทำงาน เนื่องจากการใช้ MEL เพียงไม่กี่บรรทัดสามารถช่วยทำหน้าที่ต่างๆแทนเราได้ ทีนี้เราเริ่มจากการลองเขียน MEL เพื่อสร้างต้นหญ้าต้นแรกกัน

Creating a Grass Blade

ดังที่กล่าวไว้ในตอนแรกว่าต้นหญ้ามีคุณลักษณะคล้ายกับรูปทรงกรวย (cone) ดังนั้นเราจะมาเริ่มจากการสร้างรูปทรงกรวยกันก่อน สามารถทำได้ด้วยคำสั่ง polyCone แล้วตามด้วย flags ที่ต้องการ ในที่นี้เราต้องการสร้างกรวยแบบ pyramid ที่มีสามด้านจึงต้อง flag subdivision X (-sx) = 3 รูปทรงสูงสัดส่วนระหว่าง radius (-r) กับ height (-h) จึงควรมีความแตกต่างกันมาก ในที่นี้จะตั้งค่ารัศมีเท่ากับ 0.04 และค่าความสูงเท่ากับ 1 ดังตัวอย่าง

```
polyCone
-r 0.04
-h 1
-sx 3  -sy 10  -sz 0  -axis 0 1 0
-constructionHistory off;
```

อย่าลืมว่าเราต้องสร้าง subdivisions ในแนวตั้งไว้ด้วยเพื่อความสมดุลในการดัดรูปใบ จึง flag –sy = 10 ส่วน flag axis (-ax) 0 1 0 เพื่อบอกว่าวัตถุจะถูกสร้างจากพื้นขึ้นไปบนฟ้าตามแนวแกน Y ส่วน flag สุดท้าย construction History (-ch) เป็น off เพื่อบอกให้ไม่เก็บค่า history

เมื่อสั่งสร้างวัตถุตาม scripts ที่เขียนจะพบว่าเราได้วัตถุคล้ายใบมีดจมอยู่ในแนว gridlines ขั้นต่อไปเราจะเลื่อนวัตถุของเราวางไว้บนแนว gridlines ซึ่งสามารถทำได้โดยเพิ่มคำสั่ง set attribute เข้าไปจัดการกับวัตถุ

```
      setAttr pCone1.ty 0.5;
```

เมื่อ pCone1 คือชื่อของวัตถุที่เราสร้าง แล้ว 0.5 คือครึ่งหนึ่งของความสูงวัตถุ

ก่อนที่เราจะไปในขั้นตอนต่อไป ถ้าสังเกตดูจะพบปัญหาแรกเข้ามา นั่นคือเราต้องระบุชื่อและความสูงของวัตถุในการทำงานทุกครั้ง ในการสร้างต้นหญ้าขึ้นมาซ้ำๆหลายๆต้น เราต้องตั้งชื่อและค่าให้กับมัน

ใหม่ทุกครั้งหรือ เพื่อแก้ไขปัญหานี้ ให้นักศึกษาประกาศค่าต่างๆเป็นตัวแปร floating points โดยตั้งชื่อตัวแปรค่ารัศมีว่า $grassRadius และค่าความสูงว่า $grassHeight ดังตัวอย่างด้านล่าง

```
float $grassRadius = 0.04;
float $grassHeight = 1;
```

แล้วตั้งชื่อวัตถุด้วยตัวแปร string แบบ array แทน ใช้ชื่อว่า $blades[] โดยจะตั้งให้ทันทีเมื่อสร้างวัตถุขึ้นมาดังตัวอย่างด้านล่าง

```
      string $blades[] = `polyCone
      -radius $grassRadius
      -h $grassHeight
      -sx 3 -sy 10 -sz 0 -ax 0 1 0 -ch 0ff`;
      setAttr ($blades[0] + ".ty") ($grassHeight*0.5);
```

จาก array $blades[] วัตถุใหม่ที่ถูกสร้างจะมีค่าเป็น $blades[0] ทำให้เราสามารถเลือก select ได้อย่างถูกต้อง การทำแบบนี้ไม่ใช้การเปลี่ยนชื่อของวัตถุโปรดอย่าสับสน แต่เป็นการบ่งชี้วัตถุจากลำดับของ list ของมันภายใน array นั้นๆ

เมื่อตั้งค่าต่างๆเรียบร้อยแล้ว ขั้นตอนต่อไปเราจะมาปรับขนาดของวัตถุให้มีขนาดบางคล้ายกับใบมีด โดยการ scale วัตถุเราตามแนวแกน X ลงไปเหลือ 25 เปอร์เซ็นต์ของขนาดปัจจุบัน สามารถทำได้โดยการเติมคำสั่ง set attribute นี้ลงไปต่อท้ายคำสั่งที่มีอยู่

```
      setAttr ($blades[0] + ".scaleX") 0.25;
```


951301: Advanced Animation and Rendering Techniques

เมื่อ execute คำสั่งจะได้ผลลัพธ์ดังรูปด้านล่าง

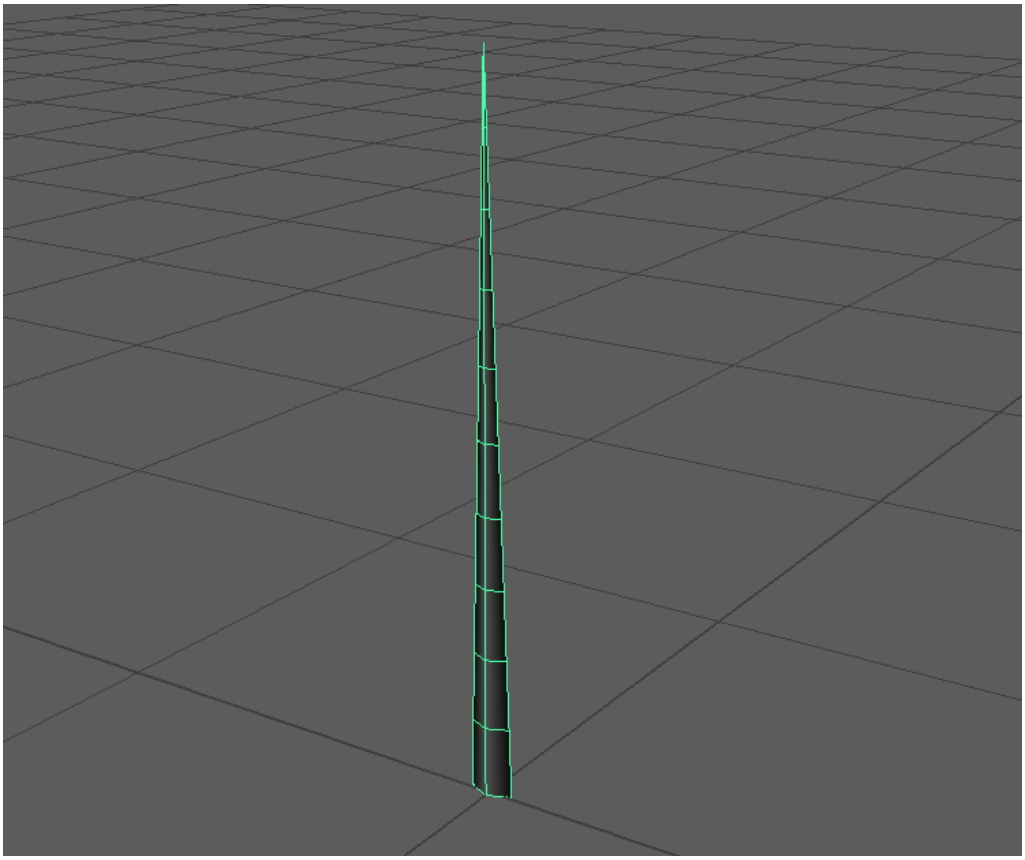

Fig 04-06: การใช้คำสั่งในการปรับแต่งคุณลักษณะให้กับต้นหญ้า

ขั้นตอนต่อไปเราจะทำการดัดใบมีดของเราให้มีลักษณะเหมือนความโค้งของใบไม้ สามารถทำได้โดยการ apply คำสั่ง deform แบบ none linear/ bend เข้าไปให้กับวัตถุที่เราสร้าง เช่นเดียวกับตัววัตถุเอง คำสั่ง bend จะต้องถูกสร้างขึ้นมาประจำวัตถุแต่ละอัน ความหมายคือเราต้องสร้างขึ้นมาเท่ากับจำนานต้นหญ้าที่จะสร้าง เราจึงควรประกาศออกมาเป็นตัวแปร string แบบ array ดังตัวอย่างด้านล่าง

```
string $bend[] = `nonLinear
                    -type bend
                    -lowBound 0
                    -highBound 2
                    -curvature -10`;
setAttr ($bend[1] + ".ty") 0;
setAttr ($bend[0] + ".envelope") 0.8;
```


951301: Advanced Animation and Rendering Techniques

เมื่อ execute คำสั่งจะได้ดังภาพตัวอย่าง

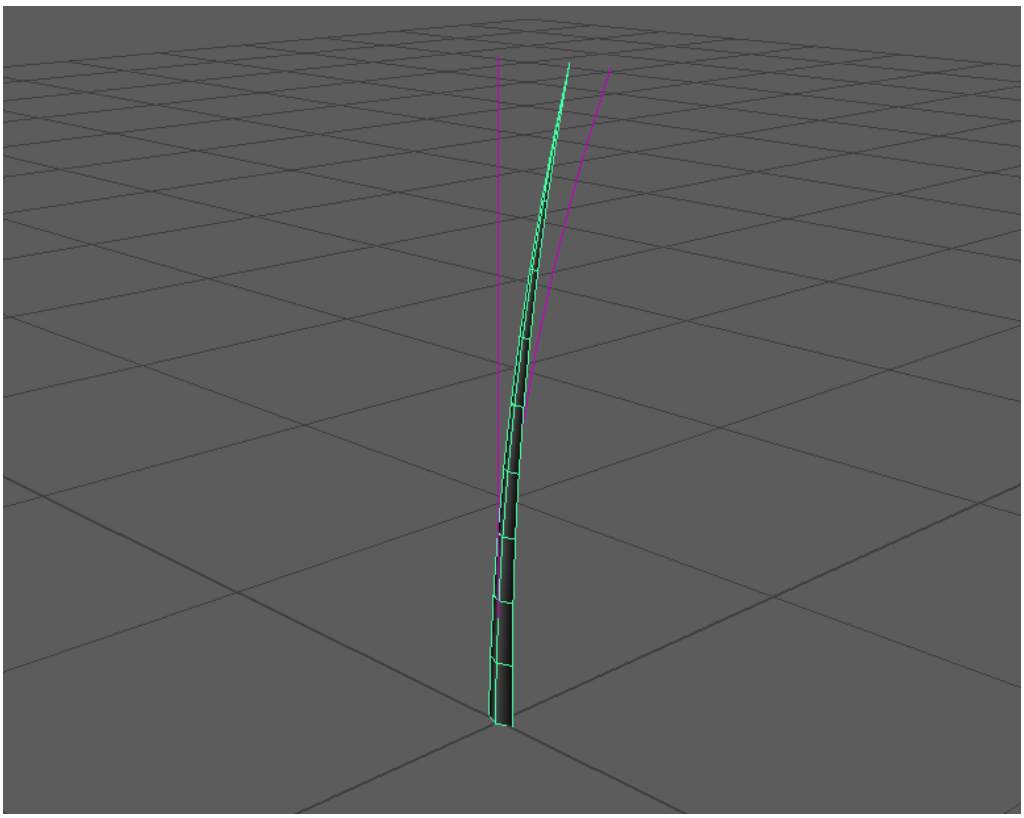

Fig 04-07: การใช้คำสั่งในการตัดลำต้นหญ้าให้มีความโค้ง

ต้นหญ้าแต่ละต้นในทุ่งหญ้าจะมีขนาด ความยาว ความโค้ง ไม่เท่ากัน ในการ generate ทุ่งหญ้าขึ้นมา เราจึงควรนำสิ่งนี้มาประยุกต์ใช้เข้าไปด้วย จากความรู้ในบทที่ผ่านมา เราได้เรียนรู้คำสั่ง random กันไปแล้ว ซึ่งเราสามารถกำหนดค่าสูงสุดกับต่ำสุดที่ใช้ในการ random ขึ้นมาได้ ในที่นี้เราจะใช้คำสั่ง random ในการสุ่มค่า attributes ต่างๆของต้นหญ้าขึ้นมาใหม่ในการ execute คำสั่งแต่ละครั้ง ให้นักศึกษาเปลี่ยนค่าตัวเลขต่างๆที่ใส่ไว้เป็นค่า random ซึ่งจะได้ชุดคำสั่งใหม่ดังด้านล่าง

```
// ประกาศค่าตัวแปรแบบ random ที่ต้องใช้ *แก้ไขของเดิม
float $grassRadius = `rand 0.02 0.06`;
float $grassHeight = `rand 0.5 3`;
float $grassCurl = `rand (-5) (-15)`;
float $envelopeVal = `rand 0.6 1.0`;

// สร้าง poly cone จากค่า random แล้ววางไว้บน gridlines *เหมือนเดิม
string $blades[] = `polyCone
-r $grassRadius
-h $grassHeight
-sx 3 -sy 10 -sz 0 -ax 0 1 0 -ch 0ff`;
setAttr ($blades[0] + ".ty") ($grassHeight*0.5);
```


951301: Advanced Animation and Rendering Techniques

```
// ปรับขนาด poly cone ให้มีลักษณะบาง *เหมือนเดิม
setAttr ($blades[0] + ".scaleX") 0.25;

// apply ค่า bend ให้กับวัตถุที่สร้าง *แก้ไข
string $bend[] = `nonLinear
                    -type bend
                    -lowBound 0
                    -highBound 2
                    -curvature $grassCurl`;
setAttr ($bend[1] + ".ty") 0;
setAttr ($bend[0] + ".envelope") $envelopeVal;
```

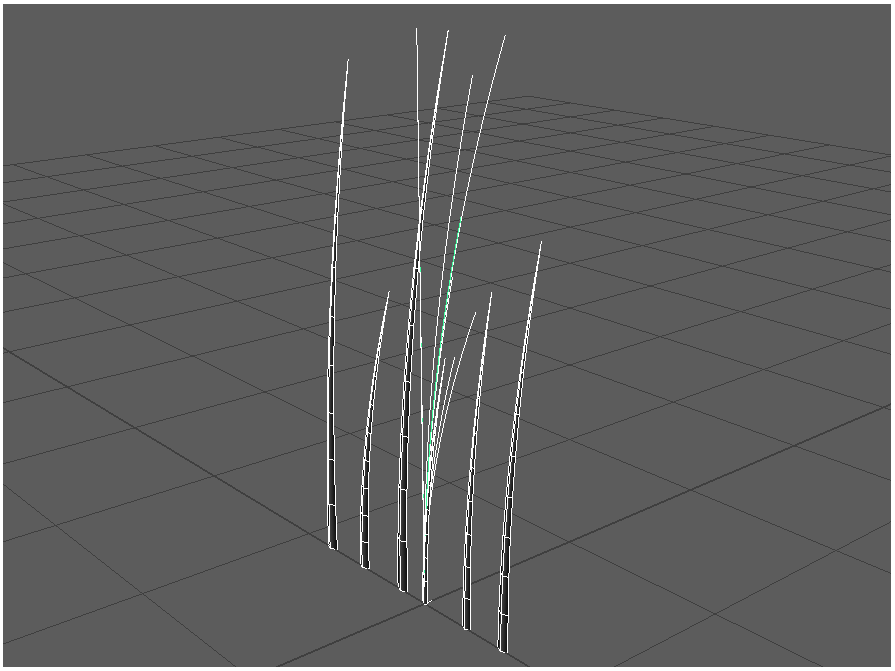

Fig 04-08: **ต้นหญ้าที่ถูกสร้างขึ้นมาโดยการสุ่มขนาดลำต้น**

เมื่อเสร็จแล้วทดลอง execute คำสั่งซ้ำๆเราจะได้ต้นหญ้าขนาดต่างๆดังภาพตัวอย่างด้านบน นักศึกษาจะสังเกตได้ว่าต้นหญ้าจะถูก generate ขึ้นมาในตำแหน่งเดียวกันตลอด เราต้องสั่ง move ไปไว้ในตำแหน่งที่เหมาะสม ขั้นตอนต่อไปเราจะสร้างการ random ตำแหน่งในการสร้างต้นหญ้าแต่ละต้นขึ้นมาเพื่อแก้ไขปัญหาในจุดนี้

ก่อนที่เราจะทำการ random ตำแหน่งของต้นหญ้า เราจะพบปัญหาว่าเมื่อลอง select เลือกที่ตัวต้นหญ้าแล้วทำการย้ายตำแหน่งของมันตามแนวแกน X และ Z จะพบว่าจะมีแต่ตังต้นหญ้าที่เคลื่อนตาม แต่ตัว bend handle ที่ใช้ควบคุมความโค้งของต้นหญ้าจะอยู่ที่ตำแหน่ง origin ซึ่งทำให้ลักษณะความโค้งของต้นหญ้าผิดรูปไป ในที่นี้ก่อนที่เราจะทำการเคลื่อนย้ายตำแหน่งของต้นหญ้า เราต้องทำ

951301: Advanced Animation and Rendering Techniques

การล้างค่า construction history (-ch) ของต้นหญ้าเสียก่อน เพื่อให้ต้นหญ้าจดจำรูปแบบของมันไว้โดยไม่ต้องใช้การคำนวณผ่าน bend handle อีกต่อไป อย่าลืมว่าภายหลังจากการล้างค่า construction history แล้ว เราจะไม่สามารถแก้ไขค่าความโค้งผ่าน bend handle ได้อีกต่อไป

ในการล้างค่า construction history สามารถทำได้ด้วยคำสั่ง delete –ch ดังตัวอย่างด้านล่าง

```
select -replace $blades;
delete -ch $blades;
```

Grasses Positioning by Random

ขั้นตอนแรกเราต้องกำหนดพื้นที่ๆเราต้องการสร้างทุ่งหญ้าขึ้นมาก่อน ในที่นี้เราจะใช้พื้นที่ของ gridline เป็นพื้นที่ของทุ่งหญ้า ขึ้นอยู่กับ version ของ Maya ที่ใช้ ในที่นี้จะยึดขนาด gridlines ของ Maya 2016 เป็นหลักซึ่งมีขนาด 12 x 12 units ดังนั้นเราจะกำหนดขนาดของการสร้างที่ X = -12 ถึง X = 12 ตามแนวแกน X และ Z = -12 ถึง Z = 12 ตามแนวแกน Z

แล้วเรายังต้องการให้ต้นหญ้าแต่ละต้นมีการหันไปในทิศทางแตกต่างกันเพื่อเลียนแบบความเป็นธรรมชาติ เราจึงใส่ค่า random ให้กับองศาที่หันของต้นหญ้าแต่ละต้น โดยสามารถทำได้โดยการ modify ค่า rotate ของตัววัตถุ scripts ที่เพิ่มเข้ามาใหม่ดังตัวอย่างด้านล่าง

```
// ประกาศตัวแปรเพื่อกำหนดขนาด กว้าง x ยาว ของสนามหญ้าที่จะสร้าง
float $minX = -12;
float $maxX = 12;
float $minZ = -12;
float $maxZ = 12;

// ประกาศตัวแปรเพื่อใช้พิกัดตำแหน่งของต้นหญ้าแบบ random ให้อยู่ภายในขอบเขตของสนามหญ้าที่สร้าง
float $randX = `rand $minX $maxX`;
float $randZ = `rand $minZ $maxZ`;
float $randRo = `rand 0 360`;



// ประกาศตัวแปรเพื่อเก็บค่าต้นหญ้าล่าสุดที่สร้างในการเรียกใช้
string $createdGrass[];
```


951301: Advanced Animation and Rendering Techniques

```
// ตั้งตำแหน่งของต้นหญ้าที่สร้าง ตามค่าที่ random ไว้ตามแนวแกน X และ Z
setAttr ($createdGrass[$i] + ".tx") $randX;
setAttr ($createdGrass[$i] + ".tz") $randZ;
setAttr ($createdGrass[$i] + ".rotateY") $randRo;
```

เมื่อ execute คำสั่งด้านบน เราจะสามารถ generate ต้นหญ้าขนาดต่างๆขึ้นในตำแหน่งภายในสนามหญ้าที่เราต้องการได้ ดังภาพตัวอย่างหลังจากการ execute คำสั่งจำนวน 10 ครั้ง จะได้ผลลัพธ์ดังนี้

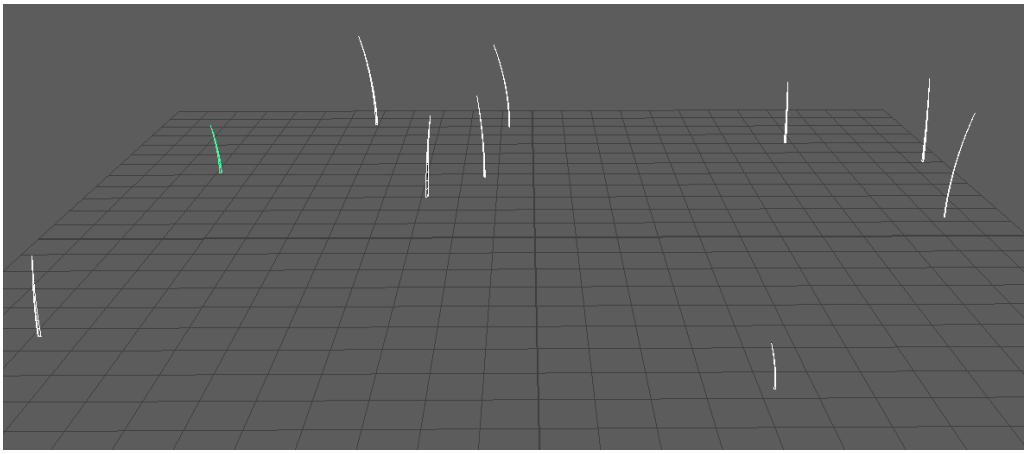

Fig 04-09: ต้นหญ้าจะถูกสร้างขึ้นมาแบบสุ่มตำแหน่งและขนาดทุกครั้งที่มีการเรียกใช้คำสั่ง

ถ้าต้นหญ้าของใครเลยออกไปนอกขอบเขต gridlines ให้ลองตรวจสอบขนาดของ gridlines ใน version ที่ใช้ ว่ามีขนาดเท่าไรแน่ แล้วแก้ไข ที่ค่า minX, maxX, minZ และ maxZ ให้สอดคล้องกัน

Generating a Grass Field

ถึงขั้นตอนนี้จะพบว่าเราสามารถ generate ต้นหญ้าแบบ random ตำแหน่งขึ้นมาได้ในทุกครั้งที่ execute คำสั่ง แต่การสร้างทุ่งหญ้าให้เต็มพื้นที่ๆแคบๆแค่ 12 x 12 units นั้น เราอาจต้อง execute คำสั่งเป็นจำนวนถึง 1000 ถึง 4000 ครั้ง ขั้นตอนต่อไปเราจะให้ Maya เป็นคนทำหน้าที่นี้ให้กับเรา โดยที่เราจะให้ Maya ตรวจสอบดูว่าต้นหญ้าได้ถูกสร้างตามจำนวนที่ต้องการหรือยัง จึงค่อยหยุดการทำงาน ขั้นตอนนี้เราสามารถใช้ for loop มาใช้

951301: Advanced Animation and Rendering Techniques

```
// ตั้งค่าความหนาแน่นของสนามหญ้า (จำนวนต้นหญ้าต่อสนาม)
int $density = 3000;
// สั่งให้ทำซ้ำจนกว่าจำนวนต้นหญ้าจะเท่ากับค่า density ที่ตั้งไว้
for ($i = 0; $i < $density; $i++)
{
        ชุดคำสั่งที่เขียนไว้ทั้งหมด
}
```

จากคำสั่งด้านบน เราประกาศค่าตัวแปรแบบ integer ขึ้นมาชื่อ $density เพื่อเก็บจำนานของต้นหญ้าที่ต้องการสร้าง จากนั้นใช้ for loop แบบเพิ่มค่าทีละหนึ่งทุกครั้งที่ run จนกว่าจะได้จำนวนต้นหญ้าตามที่ตั้งไว้ เราสามารถปรับจำนวนต้นหญ้าง่ายๆด้วยการเปลี่ยนค่า $density ในบรรทัดแรก ซึ่งจะได้ผลลัพธ์ดังนี้

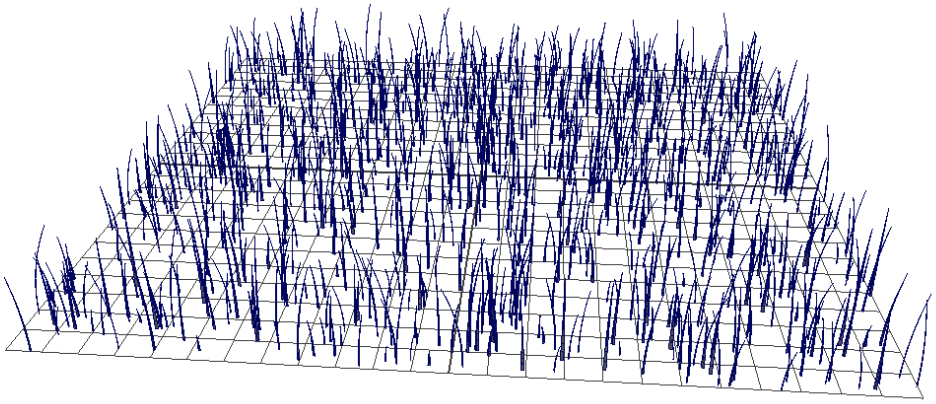

Fig 04-10: ทุ่งหญ้าที่ถูกสร้างขึ้นมาจากค่าความหนาแน่นที่ 1000

951301: Advanced Animation and Rendering Techniques

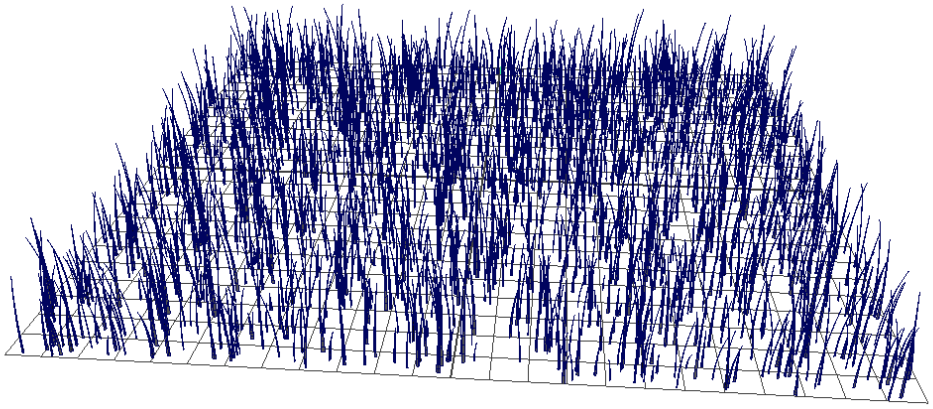

Fig 04-11: ทุ่งหญ้าที่ถูกสร้างขึ้นมาจากค่าความหนาแน่นที่ 2000

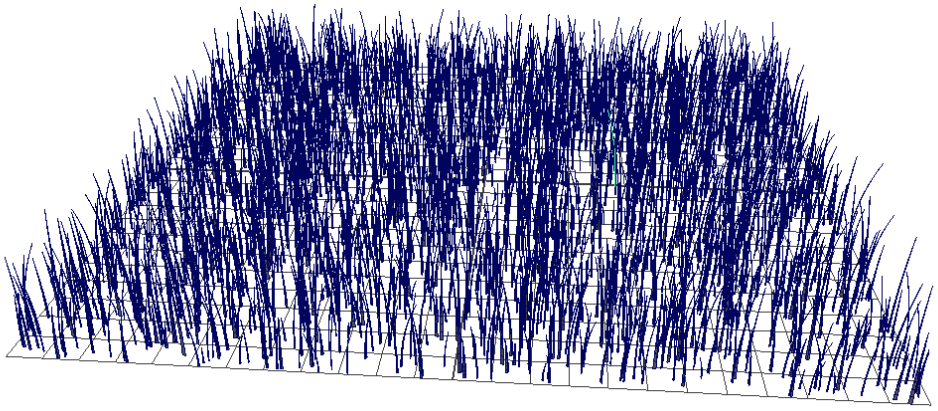

Fig 04-12: ทุ่งหญ้าที่ถูกสร้างขึ้นมาจากค่าความหนาแน่นที่ 3000

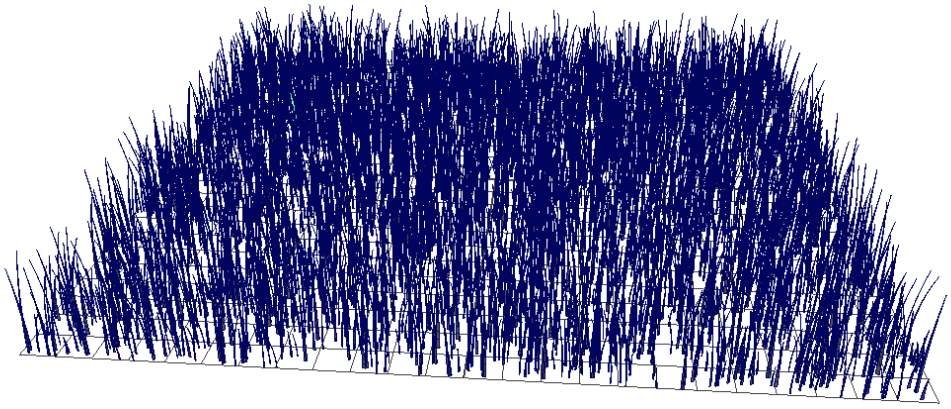

Fig 04-13: ทุ่งหญ้าที่ถูกสร้างขึ้นมาจากค่าความหนาแน่นที่ 5000

951301: Advanced Animation and Rendering Techniques

คำสั่งทั้งหมดที่ใช้ในการสร้างสนามหญ้าในบทเรียนนี้มีดังนี้

```
// set density
int $density = 100;
int $i;
for ($i = 0; $i < $density; $i++)
{
// declaration of variables
float $grassRadius = `rand 0.02 0.06`;
float $grassHeight = `rand 0.5 3`;
float $grassCurl = `rand (-5) (-15)`;
float $envelopeVal = `rand 0.6 1.0`;

// creation of polyCone
string $blades[] = `polyCone
-radius $grassRadius
-h $grassHeight
-sx 3 -sy 10 -sz 0 -ax 0 1 0 -ch 0ff`;
setAttr ($blades[0] + ".ty") ($grassHeight*0.5);

// make it thinner
setAttr ($blades[0] + ".scaleX") 0.25;

// add the bend deformer, capture the name to move it
string $bend[] = `nonLinear
                    -type bend
                    -lowBound 0
                    -highBound 2
                    -curvature $grassCurl`;
setAttr ($bend[1] + ".ty") 0;
setAttr ($bend[0] + ".envelope") $envelopeVal;
```


951301: Advanced Animation and Rendering Techniques

```
// select the cone object and delete history
select -replace $blades;
delete -ch $blades;

// set field size
float $minX = -12;
float $maxX = 12;
float $minZ = -12;
float $maxZ = 12;

// random grass creations
float $randX = `rand $minX $maxX`;
float $randZ = `rand $minZ $maxZ`;
float $randRo = `rand 0 360`;

setAttr .tx $randX;
setAttr .tz $randZ;
setAttr .rotateY $randRo;
}
```

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