เอกสารประกอบการเรียน วิชา ANI 211 การขึ้นรูปสามมิติและการออกแบบแอนนิเมชัน 1
สาขาวิชาแอนนิเมชัน ประจำวันที่ 11 และ 15 สิงหาคม ภาคการศึกษาที่ 1 ปีการศึกษา 2551
วิทยาลัยศิลปะ สื่อ และเทคโนโลยี มหาวิทยาลัยเชียงใหม่

หัวข้อ: **หลักการใส่คีย์เฟรม และการสร้างภาพเคลื่อนไหว**
(Keyframes and Timing in Animation)

วัตถุประสงค์:

1. เข้าใจถึงที่มาของ Keyframe กับ In-Between และสามารถจำแนกความแตกต่างของค่าทั้งสองในเชิงแอนนิเมชันได้
2. สามารถใช้เครื่องมือควบคุม Timeline ได้อย่างเหมาะสม
3. เรียนรู้หลักการ Set Keyframes โดยใช้ Shot Keys ความแตกต่างระหว่างการเก็บค่าเพียงลักษณะเดียว และการเก็บค่าแบบสามลักษณะพร้อมกัน
4. เรียนรู้หลักการ Set Keyframes โดยใช้หน้าต่าง Attributes Editor การตั้งค่าคีย์แบบตามหัวข้อที่ต้องการ และการตั้งคีย์แบบทั้งหมด
5. เรียนรู้หลักการ Set Keyframes โดยใช้การควบคุมแบบ Auto Keyframe Toggle ประโยชน์และความสะดวกที่ได้ เปรียบเทียบกับปัญหาที่เกิดขึ้น
6. สามารถปรับแต่งช่วงเวลา และการทำงานกับ Keyframes ทีละหลายๆคีย์พร้อมกันได้
7. การใช้ Graph Editor แก้ไขค่า In-Between ให้กับแอนนิเมชันที่สร้าง คุณสมบัติของ Tangents แต่ละประเภท รูปแบบ และเอกลักษณ์เฉพาะตัว

THE UNIVERSITY OF
CHIANGMAI
THAILAND

THE COLLEGE OF ARTS, MEDIA AND TECHNOLOGY
DEPARTMENT OF ANIMATION

ARUS KUNKHET
315, LEVEL 3, ANIMATION DEPARTMENT
THE COLLEGE OF ARTS, MEDIA AND TECHNOLOGY
THE UNIVERSITY OF CHIANGMAI 50200
THAILAND

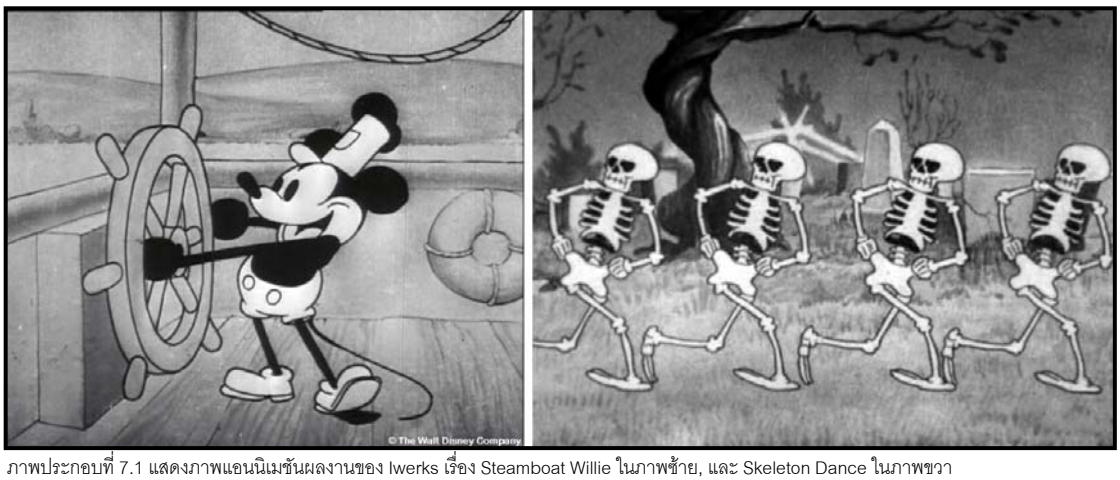
TELEPHONE +66 53 941801 (315)
FACSIMILE +66 53 893217

เนื้อหา:

เราจะสร้างแอนนิเมชันได้อย่างไร เราจะทำให้องค์ประกอบต่างๆที่สร้างขึ้นเกิดการเคลื่อนไหวได้อย่างไร เหล่านี้คือเนื้อหาที่เราจะเรียนในบทเรียนนี้ เพื่อความเข้าใจคงต้องเท้าความไปถึงการสร้างแอนนิเมชันของคุณ Walt Disney ผู้ก่อตั้งบริษัท The Walt Disney Company ซึ่งเป็นที่รู้จักอย่างกว้างขวางทั่วโลก หลายท่านอาจคิดว่าคุณ Disney เป็นนักเขียนการ์ตูน (Animator) ที่จริงแล้วงานที่เค้าทำเป็นในลักษณะผู้วางเค้าโครงเรื่องมากกว่าที่จะเป็นนักเขียนเสียเอง ส่วนผู้อยู่เบื้องหลังความสำเร็จทั้งหมดตั้งแต่เริ่มก่อตั้งบริษัทคือคุณ Ub Iwerks ซึ่งคือผู้วาดการ์ตูนของ Disney ทั้งหมด โดยทำงานในฐานะของลูกจ้างของ Disney อีกทีหนึ่ง

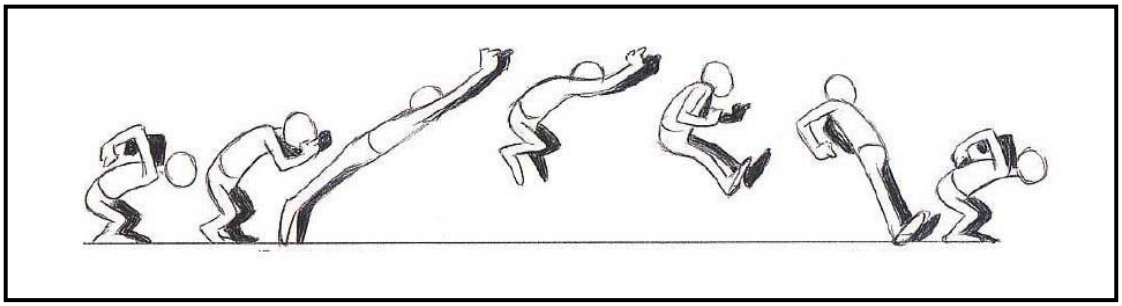
Iwerks เป็น Animator ที่มีพรสวรรค์และความละเอียดมาก อย่างที่ทราบกันว่า Frame Rate ที่ใช้ในวงการโทรทัศน์ปัจจุบันคือ 24 Frame Per Second (24 FPS) หมายความว่าในการสร้างภาพเคลื่อนไหวที่มีความยาวหนึ่งวินาที Iwerks ต้องวาดภาพทั้งหมดถึง 24 ภาพ ดังนั้นการสร้างการ์ตูนสั้นๆขนาดความยาวไม่เกิด 10 นาที จะต้องวาดภาพทั้งหมดถึง 14,400 ภาพ การสร้างภาพยนต์ขนาดยาวจึงต้องใช้ภาพทั้งหมดหนึ่งแสนถึงสองแสนภาพเลยทีเดียว

ในช่วงแรกของการก่อตั้งบริษัท Iwerks ยังสามารถจัดการกับงานทั้งหมดได้ดี และได้สร้างผลงาน Classic ไว้ประดับวงการมากมายอาทิ Steamboat Willie (ที่มีตัวเอกคือ Mickey Mouse ที่เรารู้จักกันเป็นอย่างดี) และ Skeleton Dance ทำให้บริษัทเป็นที่รู้จักอย่างกว้างขวางทั้งอเมริกา แต่เนื่องจากการเจริญเติบโตของความต้องการที่เพิ่มขึ้นตามชื่อเสียงของบริษัทที่เพิ่มขึ้น ทำให้ Iwerks ไม่สามารถสร้างงานทั้งหมดด้วยตนเองอีกต่อไป Disney จึงได้เสนอแนวความคิดใหม่ ซึ่งถือเป็นแนวทางในการสร้างแอนนิเมชันทั้งหมดในปัจจุบัน นั่นคือเรื่องของ Keyframes



ภาพประกอบที่ 7.1 แสดงภาพแอนนิเมชันผลงานของ Iwerks เรื่อง Steamboat Willie ในภาพซ้าย, และ Skeleton Dance ในภาพขวา โดยผลงานทั้งสองเรื่องนี้เป็นงานที่ไม่ได้ใช้หลักการ Keyframes แต่อย่างไร (The Walt Disney Company. 2008. All Right Received)

อะไรคือ Keyframes ในงานแอนนิเมชันยุคแรกเริ่มกล่าวคือ Disney ขอให้ Iwerks วาดเฉพาะแต่จุดสำคัญในการเคลื่อนไหวของตัวละคร ยกตัวอย่างเช่น ถ้าต้องการฉากที่ตัวละครกระโดดข้ามขอนไม้ Iwerks จะต้องวาดเพียงแค่ภาพขณะที่ตัวละครยืนอยู่หน้าขอนไม้, ตัวละครย่อตัวลงเตรียมกระโดด, ตัวละครกำลังจะลอยขึ้นจาก



ภาพประกอบที่ 7.2 แสดงจังหวะการเคลื่อนไหวหลัก หรือ Keyframes ที่ Iwerks ต้องวาด
ภาพโดยคุณฟาติมา จันทะศรี (Comgraph Co., Ltd. All Right Received)

ในการทำงานเราเรียกตำแหน่งของ Iwerks ว่า Lead Animator มีหน้าที่วาดในส่วนของการเคลื่อนไหวเฉพาะที่สำคัญๆ โดยมี Junior Animators ทำหน้าที่รับช่วงต่อ เชื่อมความสัมพันธ์ระหว่างภาพเหล่านั้น ออกมาเป็นการเคลื่อนไหวที่สมบูรณ์ เราเรียกภาพในส่วนที่ Iwerks วาดว่า *Keyframes* ในขณะที่ภาพจาก Junior Animators เรียกว่า In-Between (บางครั้งเราเรียก Junior Animator ว่า Tweener มาจากคำเต็มว่า In-Betweener นั่นเอง) ซึ่งสุดท้ายแล้ว Iwerks ไม่ได้มีความสุขกับการทำงานแบบนี้ เนื่องจากเจ้าตัวเชื่อว่าเป็นการทำลายศิลปะในงานของตัวเอง แต่นี่ไม่ใช่ประเด็นที่เราจะกล่าวถึง ประเด็นอยู่ที่แนวความคิดเรื่อง Keyframes กับ In-Betweens นี้ถูกนำมาประยุกต์ใช้ในการสร้างภาพเคลื่อนไหวด้วยคอมพิวเตอร์ต่างๆทั้งหลายในปัจจุบัน

จากโปรแกรมสร้างแอนนิเมชันต่างๆในปัจจุบัน รวมทั้ง Maya ได้ใช้แนวความคิดนี้ในการสร้างภาพเคลื่อนไหว โดยผู้ใช้มีหน้าที่เช่นเดียวกับ Iwerks นั่นคือ Lead Animator ทำหน้าที่กำหนดการเคลื่อนไหวที่สำคัญๆ (Keyframes) ในขณะที่โปรแกรมจะทำการคำนวณส่วนที่เหลือให้ (In-Between) ทำหน้าที่เสมือน Tweeners หรือ Junior Animators ให้กับเรา

เมื่อเราเข้าใจที่มาที่ไปและการทำงานของ Keyframes กับ In-Between แล้ว ก็จะสามารถตอบคำถามข้อแรกของเราว่าเราจะสร้างแอนนิเมชันได้อย่างไร แล้วแอนนิเมชันที่ดีล่ะจะสร้างได้อย่างไร ในงานสามมิติเราสามารถแบ่งองค์ประกอบของแอนนิเมชันได้เป็นสององค์ประกอบ นั่นคือวัตถุ (Objects) กับ การเคลื่อนไหว (Motions and Movements) แล้วอะไรสำคัญกว่ากันล่ะ ในที่นี้ขอยกตัวอย่างคำพูดของอาจารย์ Adam Watkins ที่ว่า “How things move is usually more important than what moves.”* ความหมายคือ เคลื่อนอย่างไรสำคัญกว่าอะไรเคลื่อน แน่นอนจากคำพูดนี้แสดงว่าการเคลื่อนไหวมีบทบาทสำคัญกว่ารูปร่างหน้าตาของสิ่งที่เคลื่อนไหวอยู่ ถ้าอย่างนั้นแปลว่า Movements สำคัญกว่า Objects เสมอไปหรือไม่ คำตอบคือไม่แน่ เพราะทั้งสององค์ประกอบจะส่งผลซึ่งกันและกัน

* Adam Watkins อาจารย์จากมหาวิทยาลัย The University of The Incarnate Word, San Antonio, Texas, USA

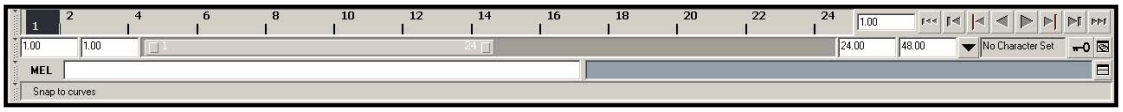
ในงานบางงานที่มีโมเดลคุณภาพต่ำ การเคลื่อนไหวที่สวยงามจะช่วยกลบเกลื่อนข้อด้อยของโมเดลได้ แต่ในทางกลับกัน มีงานอีกมากมายที่มีโมเดลคุณภาพสูงแต่กลับมีการเคลื่อนไหวที่ไม่ดี ยกตัวอย่างเช่น ตัวละครที่มีความสมจริงมากแต่กลับเคลื่อนไหวแข็งเหมือนกับหุ่นยนต์ ซึ่งจะกลบข้อเด่นของตัวโมเดลลงไปอย่างมาก หรือเรียกว่าเป็นการเคลื่อนไหวที่ทำลายงานแอนนิเมชัน

จุดที่ทำให้แอนนิเมชันต่างจากภาพนิ่งคือการเคลื่อนไหว ดังนั้นการเคลื่อนไหวจึงเป็นปัจจัยที่มีความสำคัญอย่างมากต่อภาพรวมของงานแอนนิเมชัน การสร้างการเคลื่อนไหวที่ดีช่วยให้งานแอนนิเมชันนั้นดูดีขึ้นได้ แต่การสร้างแอนนิเมชันที่ "ดี" จะต้องมีภาพที่ดีด้วย ดังนั้นทั้งสองปัจจัยจึงมีความเกื้อหนุนสอดคล้องกัน จะขาดสิ่งใดสิ่งหนึ่งไปไม่ได้

การสร้างแอนนิเมชันด้วยมายา

Creating Animation with Maya

เครื่องมือในการควบคุมและสร้างแอนนิเมชันอยู่ที่บริเวณด้านล่างของหน้าต่าง Interface ประกอบด้วยส่วนที่เป็นแถบเครื่องมือที่ใช้ในการควบคุมเวลาเรียกว่า Timeline แต่ในมายาเรียกว่า Time Slider, ส่วนที่ใช้ควบคุมการเล่นภาพเรียกว่า Playback, ส่วนควบคุมช่วงเวลาในการทำงานเรียกว่า Range Slider ดูภาพที่ 7.3 ประกอบ



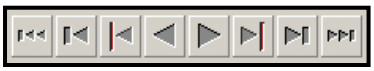
ภาพประกอบที่ 7.3 แสดง Tools ที่ใช้ควบคุมการสร้างแอนนิเมชัน บริเวณหน้าต่างด้านล่างของหน้าจอ

ที่นี้ลองมาดูรายละเอียดในแต่ละตัวว่ามีหน้าที่อย่างไร และจะสามารถควบคุมการทำงานของ Tools เหล่านี้ได้อย่างไร

Time Slider (แถบเวลา)

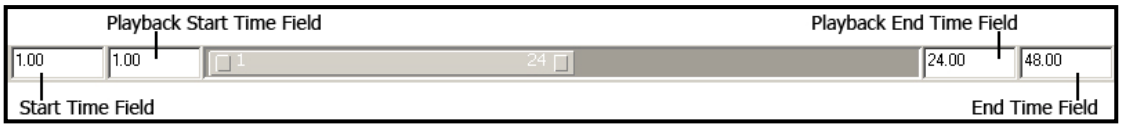
มีหน้าที่ในการควบคุมเวลา และการแสดงผลของฉากว่าจะเป็นในช่วงเฟรมใด ในส่วนนี้จะประกอบด้วยส่วนที่เป็นหมายเลขเฟรม จากภาพตัวอย่างคือแถบที่มีเลข 1 – 24 หมายเลขเหล่านี้คือหมายเลขของเฟรมนั่นเอง, ส่วนต่อมาคือส่วนที่เป็นหัวอ่าน (Current Time Indicator) ที่เป็นสี่เหลี่ยมสีดำ เป็นตัวบอกว่าภาพที่แสดงอยู่ขณะนั้นอยู่ในเฟรมใด เราสามารถลากตัวหัวอ่านนี้ไปตามแถบเฟรมเพื่อให้หน้าจอแสดงผลเฟรมที่ต้องการไล่กันไปให้เกิดเป็น Mini-Motion ในการ Preview การเคลื่อนที่ของวัตถุคร่าวๆได้ หรือจะใช้เม้าส์คลิกที่หมายเลขเฟรมใดๆเพื่อให้หัวอ่านกระโดดไปแสดงผลที่เฟรมนั้นเลยก็ได้, และส่วนสุดท้ายคือช่องสี่เหลี่ยมบริเวณขวามือ (Current Time Field) ช่องนี้จะบอกว่าเฟรมที่แสดงผลอยู่คือเฟรมอะไร เราสามารถไปที่เฟรมที่ต้องการได้โดยการใส่หมายเลขลงในช่องนี้ ตัวหัวอ่านจะกระโดดไปยังเฟรมนั้นทันที

Playback Buttons (**ปุ่มควบคุมการเล่น**)



ชุดของปุ่มควบคุมการเล่นนี้จะอยู่บริเวณด้านขวาสุดต่อจาก Time Slider ส่วนนี้จะทำหน้าที่เสมือนเป็นเครื่องเล่นภาพยนต์ที่มีปุ่ม เล่นไปข้างหน้า, เล่นย้อนกลับ, เล่นที่คีย์เฟรมถัดไป, เล่นที่คีย์เฟรมก่อนหน้า, เล่นไปข้างหน้าทีละเฟรม, เล่นย้อนกลับทีละเฟรม, เล่นไปข้างหน้าแบบเร็ว และ เล่นถอยหลังแบบเร็ว

Start/End Time Field (**จำนวนเฟรมทั้งหมดของแอนนิเมชัน**)



Start Time Field คือช่องแรกซ้ายมือที่อยู่บรรทัดเดียวกับ Range Slider ส่วน End Time Field คือช่องสุดท้ายขวามือ (อย่าสับสนกับ Playback Time Filed ที่อยู่ด้านใน) ทำหน้าที่ในการกำหนดจำนวนเฟรมทั้งหมดของแอนนิเมชันที่เราต้องการสร้าง โดยเราสามารถใส่ค่าของเฟรมแรกที่ต้องการเริ่มแอนนิเมชันได้ที่ช่อง Start Time Field และใส่ค่าของเฟรมสุดท้ายที่ End Time Field เพื่อบอกหมายเลขเฟรมสุดท้ายของงาน นักศึกษาอาจเกิดคำถามว่าแล้วทำไมต้องใส่ค่าที่ Start Time Field ให้วุ่นวายด้วย ในเมื่อแอนนิเมชันก็ควรจะเริ่มจากเฟรมที่ 1 ไม่ใช่หรือ คำตอบคือ “อาจจะใช่หรือไม่ใช่ก็ได้” ถ้าเราทำแอนนิเมชันแค่ไฟล์เดียว แต่ในความเป็นจริง การสร้างแอนนิเมชันขึ้นมาสักเรื่องหนึ่ง ต้องใช้ไฟล์มากมาย โดยแต่ละไฟล์ก็จะมีหมายเลขเฟรมต่อกันไปเรื่อยๆ เช่นไฟล์แรกเริ่มที่เฟรมที่ 1 – 100 ดังนั้นในไฟล์ที่สองเราสามารถใส่หมายเลขตรง Start Time Field ให้เริ่มที่เฟรมที่ 101 – 200 ต่อไปได้เพื่อป้องกันความสับสนว่าไฟล์ไหนมาก่อนมาหลังกัน

Playback Start/ End Time Field (**จำนวนเฟรมที่ให้แสดงผล**)

ทั้ง Playback Start และ Playback End Time Field จะอยู่ระหว่าง Start กับ End Time Field ส่วนนี้คืออะไร ในการทำงานที่มีจำนวนเฟรมมากๆ เป็นเรื่องยากที่จะเลื่อนหัวอ่านไปยังเฟรมที่ต้องการ และการมอง Keyframes บน Timeline ก็ไม่สามารถทำได้อย่างสะดวก Playback Start/ End Time Field จะช่วยให้เราสามารถขยาย Timeline ออกเป็นช่วงๆ ปิดการแสดงผลของเฟรมที่ไม่ต้องการ ทำให้เกิดที่ว่างบน Timeline เพิ่มขึ้น ทำให้การทำงานเกิดความสะดวกสบายขึ้น โดยจะไม่มีผลกับจำนวนเฟรมแท้จริงในการประมวลผลแต่อย่างใด เราสามารถใส่ค่าช่วงของเฟรมที่ต้องการขยายได้โดยใส่หมายเลขเฟรมแรกที่ช่อง Playback Start Time Field และเฟรมสุดท้ายที่ Playback End Time Field โดยช่วงของเฟรมที่ใส่ไม่สามารถทะลุออกจากช่วงของเฟรมที่ใส่ใน Start/ End Time Field ได้ (ถ้าเกินโปรแกรมจะเปลี่ยนเวลาใน Start กับ End Time Field ให้ตรงกัน)

Range Slider (**แถบเลื่อนระยะของเฟรมที่แสดงผล**)

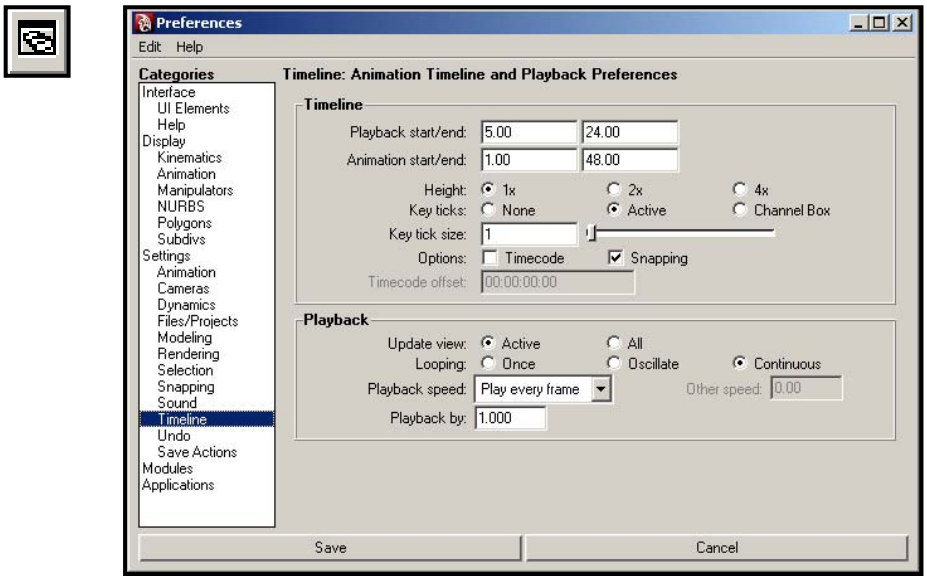
แถบเลื่อนระยะนี้ถ้าสังเกตให้ดีจะมีหมายเลขของเฟรมอยู่ที่บริเวณหัวกับท้ายของแถบ แสดงหมายเลขของเฟรมแรกและเฟรมสุดท้ายของช่วงเวลา ซึ่งจะตรงกับหมายเลขที่อยู่ภายในช่อง Playback Start/ End Time Field เรา

Auto Keyframe Toggle (**ปุ่มบันทึกคีย์เฟรมแบบอัตโนมัติ**)

ปุ่มนี้จะอยู่บริเวณด้านขวาของ Range Slider สามารถเปิดการทำงานได้โดยการคลิกที่ปุ่มโดยตรง ปุ่มจะเปลี่ยนเป็นสีแดงดังภาพตัวอย่าง แสดงว่าการทำงานของ Auto Keyframe Toggle ได้ถูกเปิดขึ้นแล้ว Auto Keyframe Toggle มีประโยชน์อย่างไร เมื่อเราเคลื่อนไหววัตถุใดๆภายในฉาก โปรแกรมจะทำการเก็บค่าเหล่านั้นไว้ตั้งเป็น Keyframes ให้โดยอัตโนมัติ โดยที่เราไม่ต้องทำการ Set Keys แต่อย่างใด ประโยชน์คือเรื่องของความสะดวกสบายในการทำงาน แต่ข้อเสียคือเราไม่สามารถกำหนดค่าการเคลื่อนไหวเฉพาะบางอย่างได้ และการทำงานอาจเกิดข้อผิดพลาดโดยไม่ตั้งใจ สำหรับผู้ใช้ที่ยังขาดความชำนาญในเรื่องของการตั้งค่า Keyframe ได้

หมายเหตุ: การทำงานของ Auto Keyframe Toggle จะสามารถเริ่มได้ จะต้องมีอย่างน้อย 1 คีย์เฟรมที่เรา Set เข้าไปด้วยวิธีแบบปกติก่อน

Animation Preferences Button



ปุ่มนี้จะอยู่ถัดไปจากปุ่ม Auto Keyframe Toggle มีหน้าที่ควบคุม Preferences ของ Timeline เมื่อเรากดที่ปุ่มนี้ หน้าต่าง Preferences ในหัวข้อของ Timeline จะถูกเปิดขึ้นมา (หรือสามารถเปิดได้โดยไปที่ **Window/ Setting/Preferences/ Preferences** แล้วไปที่หัวข้อ **Timeline** ได้เช่นกัน) ในหน้าต่างๆนี้แบ่งเป็นสองหมวดใหญ่ๆคือ Timeline กับ Playback ในหมวด Timeline เราสามารถแก้ไขค่าต่างๆที่เราตั้งใน Timeline ได้จากส่วน

นี้ ซึ่งจะได้ผลลัพธ์เช่นเดียวกัน ส่วนที่สองคือ Playback จะเป็นตัวควบคุมการเล่นของ Playback Buttons ซึ่งมีหัวข้อต่างๆเช่น

หัวข้อ **Looping** จะเป็นตัวกำหนดว่าเราจะ Preview ในแบบเล่นครั้งเดียวแล้วหยุด (Once), เล่นแบบไปแล้วย้อนกลับ (Oscillate), หรือแบบวนเป็น Loop (Continuous)

หัวข้อ **Playback Speed** เป็นการกำหนดความเร็วในการเล่น ประกอบด้วย **Play Every Frame** เครื่องจะเล่นโดยใช้ความเร็วของ CPU กล่าวคือยิ่งเครื่องมีการประมวลผลที่เร็วเท่าไหร่ ภาพจะถูกเล่นเร็วขึ้นเท่านั้น โดยไม่ใช่ความเร็วที่แท้จริงของภาพยนต์ตัวนั้น (ไม่ใช่ 24 FPS), **Real-Time (24 FPS)** เครื่องจะเล่นโดยใช้ความเร็วจริงของภาพยนต์ คือ 24 เฟรมต่อวินาที การเลือกหัวข้อนี้จะทำให้เราสามารถทราบ Timing ที่แท้จริงของงานได้, **Half (12 FPS)** เครื่องจะเล่นโดยหน่วงความเร็วลงครึ่งหนึ่ง หรือคือการเล่นแบบ Slow Motion ลงที่ครึ่งหนึ่งของความเร็วปกติ, **Twice (48 FPS)** จะเล่นที่ความเร็วเป็นสองเท่าจากความเร็วปกติ

หัวข้อ **Playback By** ค่าตั้งต้นจะอยู่ที่ 1.00 นั่นคือเครื่องจะเล่นทุกๆเฟรม ถ้าใส่ค่าอื่นลงไปเช่น 5 เครื่องจะเล่นทุกๆห้าเฟรมเป็นต้น

การตั้งค่าคีย์เฟรม (Keyframe Settings)

การตั้งค่าคีย์เฟรมสามารถทำได้หลายวิธีไม่ว่าจะเป็นการใช้คีย์ลัด (Short Keys), การคลิกขวาที่หน้าต่าง Attributes ของวัตถุ, หรือแม้กระทั่งการเปิด Auto Keyframe Toggle ให้โปรแกรมใส่คีย์เฟรมให้แบบอัตโนมัติ ดังนั้นก่อนที่เราจะมาเรียนรู้การสร้างแอนนิเมชัน เราควรทำความเข้าใจกับวิธีการใส่คีย์เฟรมด้วยวิธีต่างๆก่อน

การใส่คีย์เฟรม (Insert Keyframes)

การใส่คีย์เฟรมโดยใช้คีย์ลัด

การใช้คีย์ลัดทำให้การทำงานเกิดความสะดวกสบายขึ้นมาก ก่อนอื่นเรามาทำความรู้จักกับคีย์ลัดเหล่านี้ก่อนว่าคีย์ไหนทำหน้าที่อะไรบ้าง

- **Shift+W** คือการใส่ค่าคีย์เฟรมให้กับวัตถุในตำแหน่งที่อยู่ขณะนั้น **(Position)** โดยคำนวณจากตำแหน่งในแนวแกน X, Y และ Z โปรแกรมจะใส่ค่าความเปลี่ยนแปลงให้กับค่า Translate X, Y และ Z
- **Shift+E** คือการใส่คีย์เฟรมให้กับตำแหน่งการหมุนของวัตถุ **(Rotation)** เช่นหมุนไปจากเดิมกี่รอบ ในทิศทางไหนเป็นต้น โปรแกรมจะใส่ค่าความเปลี่ยนแปลงให้กับค่า Rotate X, Y และ Z
- **Shift+R** คือการใส่คีย์เฟรมให้กับการเปลี่ยนขนาดของวัตถุ **(Scale)** เช่นวัตถุมีการขยาย การหดตัวจากเดิมอย่างไร โปรแกรมจะใส่ค่าความเปลี่ยนแปลงให้กับ Scale X, Y และ Z

จะสังเกตได้ว่าในคีย์ลัดแต่ละตัวจะจดจำเฉพาะค่าเฉพาะตัวแต่ละอย่าง เช่น *Shift + W* จะจำแค่ตำแหน่งของวัตถุ โดยที่ไม่จำค่าของ Rotation หรือ Scale แต่อย่างใด หมายความว่าถ้าเราเลื่อนตำแหน่งของวัตถุ ในขณะเดียวกันเราก็หมุนวัตถุรอบตัวเองสามรอบด้วย Rotate Tool และทำการ Scale วัตถุให้มีขนาดใหญ่ขึ้น แล้ว

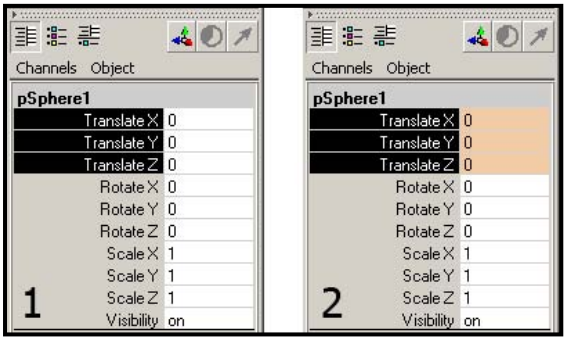
ในการทำงานที่ไม่ต้องการแยกการบันทึกค่าคีย์เฟรมทั้งสามออกจากกัน เราสามารถทำการรวมค่าการเปลี่ยนแปลงเหล่านั้นไว้ที่การกดคีย์ลัดเพียงปุ่มเดียวนั่นคือ “S”

- **กดปุ่ม S** คือการสั่งให้โปรแกรมใส่ค่าคีย์เฟรมทั้งสามคือ Move, Rotate และ Scale ลงไปที่เฟรมนั้น ไม่ว่าวัตถุจะมีการเปลี่ยนแปลงในหัวข้อนั้นหรือไม่ คีย์เฟรมก็จะถูกติดตั้งลงไป ข้อเสียคืออาจก่อให้เกิดคีย์เฟรมที่ถูกสร้างขึ้นมาโดยไม่ได้ใช้ได้ ทำให้กินการประมวลผลของเครื่อง ดังนั้นควรเลือกใช้ให้ดีว่าคีย์ไหนเหมาะกับสถานการณ์อย่างไร

ภายหลังจากที่เรากดคีย์ลัดแล้ว เราสามรถตรวจสอบดูว่าคีย์เฟรมถูกบันทึกลงไปแล้วหรือไม่ โดยให้สังเกตจากเส้นสีแดงที่จะถูกสร้างขึ้นบน Timeline เป็นตัวแสดงว่าค่าคีย์เฟรมได้ถูกทำการบันทึกไว้แล้ว

การใส่คีย์เฟรมโดยใช้วิธี Attributes Editor

การใส่คีย์เฟรมโดยการคลิกขวาที่ Attributes Editor นี้จะสามารถให้ค่าที่ละเอียดกว่า คือนอกจากจะแยกการใส่คีย์เฟรมออกเป็นสามส่วนคือ Translate, Rotate และ Scale แล้ว ยังสามารถแยกย่อยลงไปกว่านั้นได้เช่นใส่ค่า Translate X, Y และ Z แยกจากกันเป็นต้น ดังนั้นจึงเหมาะกับการทำงานที่ต้องการความละเอียดสูง แต่ข้อเสียคือขั้นตอนการใส่คีย์เฟรมที่มีความยุ่งยากกว่านั่นเอง คือแทนที่จะสามารถกดคีย์ลัดเพื่อทำการบันทึกได้เลย เราต้องไปตั้งค่าที่หน้าต่าง Attribute Editor ของวัตถุนั้นก่อน ซึ่งมีขั้นตอนดังนี้



ภาพประกอบที่ 7.4 แสดงการใส่คีย์เฟรมใน Attributes Editor

1. เมื่อ Select เลือกวัตถุอยู่ ที่แถบ Attributes Editor ของวัตถุนั้น บริเวณด้านบนที่แสดงค่า Translate, Rotate และ Scale ให้คลิกเลือกในหัวข้อที่ต้องการใส่คีย์เฟรม (คลิกในบริเวณที่เป็นสีเทา จะเกิดเป็นแถบสีดำขึ้นมา) ถ้าต้องการเลือกมากกว่าหนึ่งหัวข้อ สามารถทำได้โดยการกด Shift ค้างไว้ แล้วเลือกหัวข้อที่เหลือ จากภาพตัวอย่างนี้เลือกสามหัวข้อ
2. เมื่อทำแถบสีดำ Highlight หัวข้อที่ต้องการแล้ว ให้คลิกขวาที่บริเวณแถบสีดำนั้น แล้วเลือกไปที่ **Key Selected** เพื่อบอกกับโปรแกรมให้ใส่คีย์เฟรมลงในหัวข้อที่เลือก ผลลัพธ์ที่ได้คือบริเวณช่องสีขาวด้านหลังหัวข้อ (บริเวณที่ใส่เลข) จะเปลี่ยนเป็นสีส้ม แสดงว่าคีย์เฟรมถูกบันทึกลงในหัวข้อนั้นๆเรียบร้อยแล้ว แถบที่เป็นสีส้มนี้จะไม่สามารถเปลี่ยนแปลงค่าด้วยการพิมพ์ตัวเลขได้ จนกว่าจะมีการ Set Key เข้าไปใหม่

หลังจากขั้นตอนนี้แล้ว เราสามารถเลื่อนหัวอ่านไปยังเฟรมถัดๆไป จัดการกับวัตถุให้อยู่ในตำแหน่ง ทิศทาง และขนาดที่ต้องการ แล้วทำตามขั้นตอนที่ 1 และ 2 ต่อไป เพื่อใส่คีย์เฟรมให้กับเฟรมอื่นๆที่ต้องการ

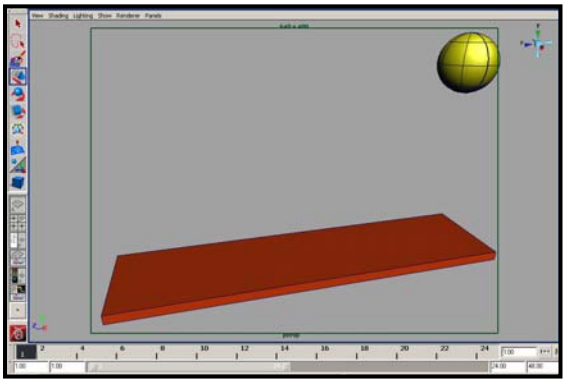
หมายเหตุ: ถ้าต้องการใส่คีย์เฟรมให้กับหัวข้อทั้งหมด (Translate, Rotate และ Scale) สามารถทำได้โดยการคลิกขวาที่หัวข้อใดๆใน Attributes Editor แล้วเลือก ***Key All*** *จะเกิดแถบสีส้มที่ทุกหัวข้อ ซึ่งมีค่าเท่ากับการกดคีย์ลัด "S" นั่นเอง*

การใส่คีย์เฟรมด้วยวิธี Auto Keyframe Toggle

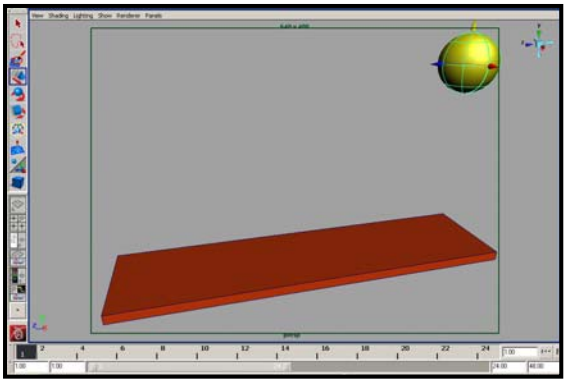
วิธีนี้สามารถทำได้ง่าย และมีความสะดวกอย่างมากสำหรับงานที่ไม่ต้องการความละเอียดหรือซับซ้อนเท่าไร มีขั้นตอนคือเราต้องใส่คีย์เฟรมให้กับวัตถุด้วยวิธีปกติก่อนอย่างน้อยหนึ่งคีย์เฟรม (จะเป็นการใช้คีย์ลัดหรือการใส่จาก Attributes Editor ก็ได้) หลังจากนั้นทำการเปิดการทำงานของ Auto Keyframe Toggle โดยคลิกที่ไอคอนรูปกุญแจให้เกิดเป็นสีแดง แล้วเลื่อนหัวอ่านไปยังเฟรมที่ต้องการ แล้วจัดการความเปลี่ยนแปลงให้กับวัตถุได้เลย เช่น ย้ายที่ หมุน หรือย่อ/ขยายขนาด ที่ Timeline จะเกิดขีดสีแดงของคีย์เฟรมให้โดยอัตโนมัติในทุกๆครั้งที่เราสร้างความเปลี่ยนแปลงให้กับวัตถุ

ทดลองทำลูกบอลกระเด้งกระดอน
(Bouncing Ball)

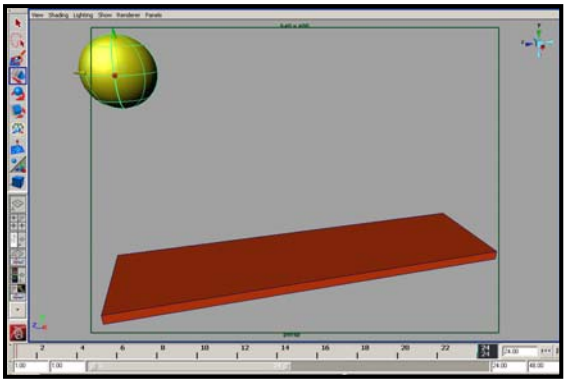
เนื่องจากการสร้างแอนนิเมชันเป็นเรื่องที่ซับซ้อนและต้องอาศัยความเข้าใจในการทำ จึงเป็นสิ่งจำเป็นที่เราจะต้องทดลองทำด้วยตนเองให้เกิดความเข้าใจ ในหัวข้อนี้จึงจะเสนอแนะขั้นตอนการสร้างแอนนิเมชันสุด Classic คือการสร้างลูกบอลกระเด้งกระดอนพื้น โดยที่จะมีการยุบและยืดของลูกบอลในขณะที่กระทบพื้นด้วย มีขั้นตอนดังนี้



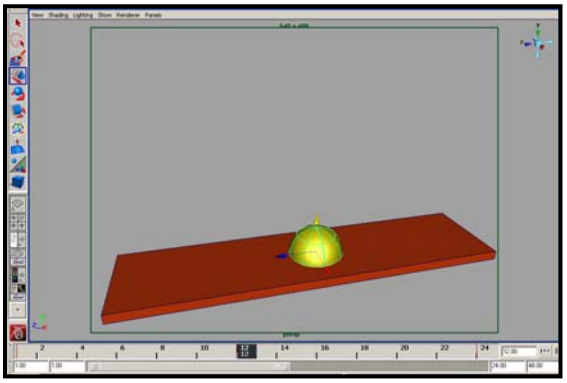
1. ให้สร้างฉากขึ้นมาประกอบด้วยลูก Sphere หนึ่งลูกจาก NURBS และพื้นสำหรับให้ลูกบอลเด้งอีกหนึ่งอันโดยใช้ Polygons Cube จัดวางให้อยู่ตำแหน่งตามภาพตัวอย่าง โดยที่ให้ลูกบอลลอยสูงจากพื้นพอสมควร เนื่องจากเรากำลังจะสร้างฉากที่ลูกบอลตกกระเด้งลงมาที่พื้นก่อนที่จะกระเด้งลอยออกจากฉากไป



2. ดูให้แน่ใจว่าหัวอ่านอยู่ที่เฟรมแรก จากนั้น Select ที่ลูกบอล แล้วกดปุ่ม “S” เพื่อใส่คีย์เฟรมให้กับลูกบอลที่ลอยอยู่กลางอากาศ สังเกตที่ Timeline ตรงเฟรมที่ 1 จะมีขีดสีแดงปรากฏขึ้น แสดงว่าคีย์เฟรมถูกใส่ลงไปเรียบร้อยแล้ว



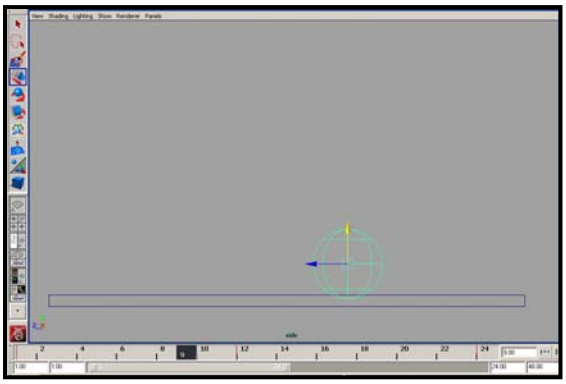
3. เลื่อนหัวอ่านไปยังเฟรมที่ 24 แล้วใช้ Move Tool เคลื่อนย้ายลูกบอลไปทางซ้ายของหน้าจอ เสร็จแล้วกดปุ่ม “S” อีกครั้งเพื่อทำการบันทึกค่าความเปลี่ยนแปลงที่เฟรมที่ 24 นี้ จากนั้นตรวจสอบบน Timeline อีกครั้งว่ามีขีดสีแดงขึ้นที่เฟรมนี้หรือไม่



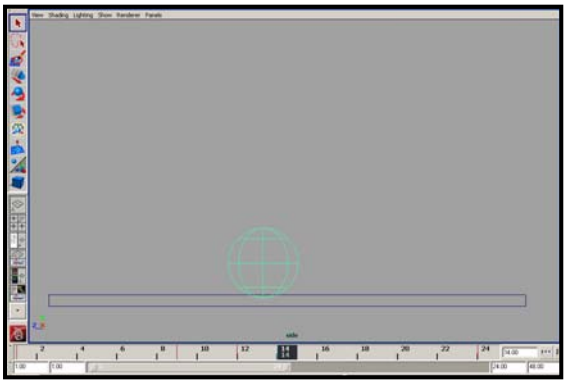
4. เลื่อนหัวอ่านกลับมายังเฟรมที่ 12 ลูกบอลจะวิ่งกลับมาครึ่งระยะทาง ไม่ต้องสนใจ ให้ใช้ Move Tool ดึงลูกบอลลงมาที่พื้นในแนวดิ่ง คือให้ดึงลงมาตรงๆ อย่าให้เฉไปทางด้านหน้าหรือด้านข้าง โดยให้ลูกบอลจมลงไปใต้พื้นครึ่งใบ แล้วกด “S” เพื่อใส่คีย์เฟรมที่สามลงไป

จากตรงนี้ถ้าเราทดลองกด Play จาก Playback จะพบว่าลูกบอลลอยอยู่แล้วตกกระทบพื้น จมลงไปครึ่งลูก แล้วจึงเด้งขึ้นมา มีลักษณะการวิ่งเป็นรูปตัว V ดูแล้วให้ความรู้สึกที่แข็งๆหลอกตามาก และลูกบอกก็ยังดูไม่เหมือนว่าเด้งขึ้นมาเมื่อกระทบพื้น แต่เหมือนลอยอยู่แบบไร้น้ำหนักมากกว่า ทั้งนี้เนื่องจากเรายังไม่ได้ใส่หลักการเรื่องของ Squash and Stretch (การยุบและยืดตัวของวัตถุ) ให้กับลูกบอล หลักการนี้เป็นปัจจัยที่สามารถ Apply ลงไปได้ในแอนนิเมชันเกือบทุกประเภท ไม่ว่าจะเป็นลูกบอลที่ตกกระทบพื้น ลูกเบสบอลที่ถูกไม้ตี ต้นไม้ที่ถูกลมพัด ก้อนเยลลี่ (น่าทาน) ที่สั่นเนื่องจากอยู่ในจานที่มีการเคลื่อนไหว หรือแม้กระทั่งการกระโดดข้ามสิ่งกีดขวางของคน ก็ต้องใช้หลักการนี้เช่นกัน ดูภาพประกอบที่ 7.2 ก่อนที่คนจะกระโดดขึ้นต้องมีการย่อขาก่อน เป็นการยุบตัวลงไปจากจุดที่ยืนอยู่ปกติเพื่อเป็นการเพิ่มแรงส่ง ก่อนที่จะทำการกระโดด ในจังหวะนี้เรียกว่า Squash และในจังหวะที่

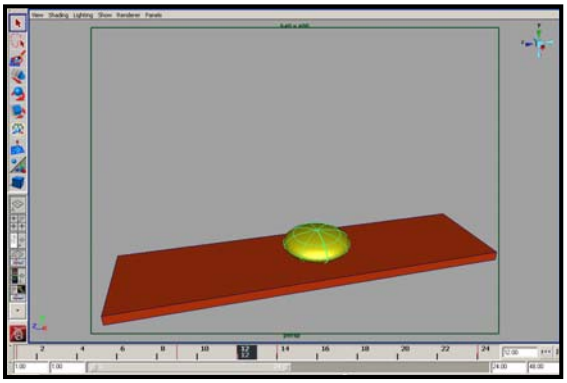
กลับมาที่เรื่องของลูกบอล เมื่อลูกบอลสัมผัสจะเกิดการยุบตัวจากแรงกระแทก (Squash) ก่อนที่จะมีการยืดตัวออกในจังหวะที่เด้งออกจากพื้น เมื่อเราเข้าใจสภาพความเป็นจริงของลูกบอลแล้ว ก็จะสามารถสร้างการเคลื่อนไหวที่สมจริงขึ้นให้กับงานได้ รายละเอียดเล็กๆน้อยๆเหล่านี้ที่ หลายๆคนมองข้ามไป เป็นตัวสร้างความแตกต่างระหว่างแอนนิเมชันธรรมดาๆ กับงานที่มีความเป็นเอกลักษณ์ได้
ดังนั้นเราลองมาดูว่าเราจะสร้างสิ่งเหล่านี้ได้อย่างไร



5. กลับมาที่มุมมอง Side View จะพบว่าในเฟรมที่ลูกบอลเริ่มที่จะสัมผัสพื้นคือเฟรมที่ 9 และเฟรมที่ลูกบอลกำลังจะลอยพ้นพื้นคือเฟรมที่ 15 เหตุการณ์ที่ลูกบอลจะเกิดการเปลี่ยนแปลง เริ่มจากยุบตัวลงจากแรงกระแทก ก่อนที่ละกลับมาสู่สภาพเดิมและลอยพ้นพื้นไป ล้วนเกิดขึ้นในระหว่างสองเฟรมนี้ (เฟรมที่ 9 - 15)

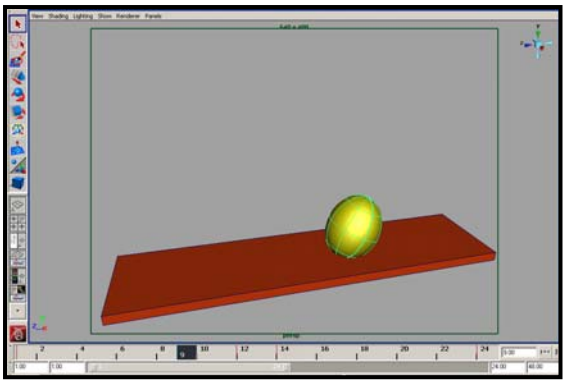


6. เลื่อนหัวอ่านไปยังเฟรมที่ 9 แล้วทำการใส่คีย์เฟรมลงไปโดยการกด “S” จากนั้นเลื่อนไปยังเฟรมที่ 14 แล้วทำการใส่คีย์เฟรมโดยการกด “S” อีกทีหนึ่ง (ที่ไม่ใส่ในเฟรมที่ 15 เนื่องจากต้องมีจังหวะที่ลูกบอลเหยียดตัวออก ที่เราจะทำต่อไป) ในขณะนี้เราจะมีคีย์เฟรมทั้งหมดอยู่ 5 อัน คือในเฟรมที่ 1, 9, 12, 14 และ 24 ตามลำดับ

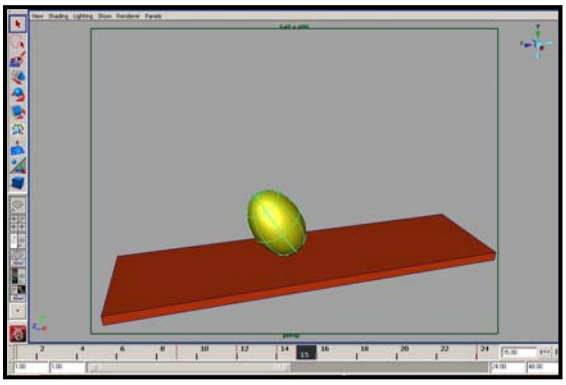


7. เลื่อนหัวอ่านกลับไปที่เฟรมที่ 12 แล้วใช้ Scale Tool ทำการบี้ลูกบอลให้แบนลง แล้วขยายขนาดของวงกลมให้ใหญ่ขึ้น มีลักษณะเหมือนลูกบอลลมที่ถูกอัดลงพื้นอย่างแรง เมื่อได้รูปทรงที่ต้องการแล้ว ให้ใช้ Move Tool ทำการยกลูกบอลให้วางอยู่บนพื้นพอดิบพอดี อย่าให้จมไปใต้พื้น จัดให้อยู่ตามภาพตัวอย่าง

ลอง Play Movie ดูจะพบว่าลูกบอลมีการยุบตัวลงเมื่อตกกระทบพื้น ทำให้เห็นถึงแรงกระแทกได้ดีขึ้น มีลักษณะเหมือนลูกบอลยางซึ่งถือเป็นการสิ้นสุดของขั้นตอน Squash ในขั้นตอนต่อไปเราจะสร้างการยืดตัวในจังหวะที่ลูกบอลกระดอนขึ้นจากพื้น

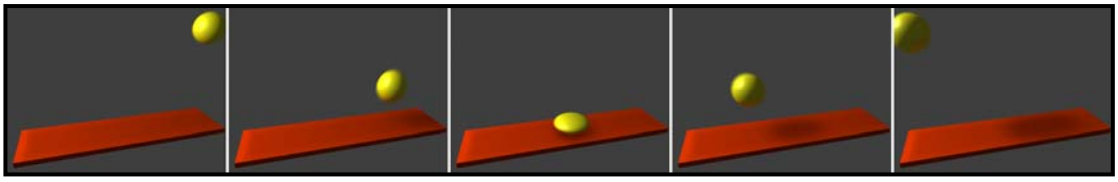


8. เลื่อนหัวอ่านกลับไปที่เฟรมที่ 9 ใช้ Scale Tool ยืดความยาวลูกบอลออก แล้วลดขนาดเส้นรอบวงของลูกบอลลง จากนั้นใช้ Rotate Tool หมุนลูกบอลให้หันไปในทิศทางเสมือนว่ามันกำลังพุ่งอยู่ตามภาพตัวอย่าง เมื่อได้ตามต้องการแล้วให้กด “S” เพื่อใส่ค่าคีย์เฟรมลงไป



9. ไปยังเฟรมที่ 15 ทำเช่นเดียวกับข้อ 8 เพียงแต่ให้หันทิศทางการเด้งของลูกบอลตรงข้ามกัน แล้วใส่ค่าคีย์เฟรมลงไปด้วยปุ่ม “S” อีกทีหนึ่ง เมื่อเสร็จขั้นตอนนี้แล้วเราจะมีคีย์เฟรมทั้งหมด 6 อัน คือที่เฟรมที่ 1, 9, 12, 14, 15 และ 24 ตามลำดับ

ลอง Play Movie ดูอีกครั้ง จะพบว่าลูกบอลที่มีการใส่ค่า Squash และ Stretch ลงไปจะดูมีความสมจริงเพิ่มขึ้น ดูภาพประกอบที่ 7.5 สังเกตการเคลื่อนที่ของวัตถุ อย่าลืมปรับอัตราการเล่นให้เป็นแบบ Real-Time (24 FPS) ก่อน เพื่อให้เห็นความเร็วที่แท้จริงขิงการเล่น (สามารถทำได้โดยการกดปุ่ม Animation Preferences บริเวณด้านขวามือของ Range Slider)



ภาพประกอบที่ 7.5 แสดงการกระเด้งของลูกบอล จากการตั้งค่าที่ได้

จากการ Preview ดูจะพบว่าลูกบอลมีการเคลื่อนที่ๆออกจะเร็วๆไปอยู่สักนิด ซึ่งการปรับแอนนิเมชันที่มีการ Set Keyframes ไปแล้ว ให้เร็วหรือช้าลงนั้น สามารถทำได้จากการปรับแต่งส่วนของ Timeline ซึ่งรานละเอียดจะอยู่ภายในหัวข้อถัดไป

การปรับแต่งช่วงเวลา

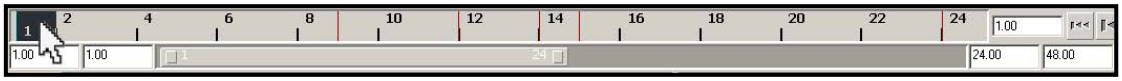
Timeline Editing

เราสามารถทำการแก้ไขคีย์เฟรมที่ทำการตั้งค่าไปแล้วได้โดยการคลิกขวาที่ Timeline จะปรากฏเมนูมาให้เราเลือก ไม่ว่าจะเป็นคำสั่ง Copy, Cut, Delete, Paste ก็สามารถทำกับคีย์เฟรมได้ไม่ต่างกับการใช้คำสั่งเหล่านี้กับโปรแกรมอื่นๆ ยกตัวอย่างเช่นถ้าเราสร้างคีย์เฟรมใดๆที่ไม่ต้องการขึ้นมา เราสามารถลบทิ้งได้โดยการเลื่อนหัวอ่านไปยังเฟรมนั้น คลิกขวาแล้วเลือก Delete คีย์เฟรมนั้นจะถูกลบทิ้งไป หรือในกรณีที่เราต้องการสร้างคีย์เฟรมสองอันที่มีค่าเท่ากันทุกอย่าง เช่น ในเฟรมที่ 1 ลูกบอลอยู่บนพื้น จากนั้นลูกบอลลอยขึ้นแล้วตกลงมาในจุดเริ่มต้นอีกครั้งในเฟรมที่ 12 เราสามารถเลื่อนหัวอ่านมาที่เฟรมที่ 1 แล้วทำการ Copy คีย์เฟรม จากนั้นไปยังเฟรมที่ 12 แล้วเลือก Paste เพื่อให้สองคีย์เฟรมนี้มีค่าเท่ากันทุกอย่างเป็นต้น

ที่กล่าวมาถึงนี้เป็นการจัดการกับคีย์เฟรมทีละคีย์เฟรม ในกรณีที่เราต้องการย้ายคีย์เฟรมพร้อมๆกันทีละหลายๆคีย์เฟรมล่ะ จะสามารถทำได้อย่างไร จากเรื่อง Bouncing Ball ในหัวข้อที่ผ่านมา แอนนิเมชันเริ่มในเฟรมที่ 1 และจบลงในเฟรมที่ 24 ถ้าเราต้องการให้แอนนิ้มชันเริ่มช้าไปห้าเฟรม (เริ่มในเฟรมที่ 5 จบในเฟรมที่ 29) จะทำการย้ายคีย์เฟรมทั้งชุดเลื่อนถอยไปพร้อมกันได้อย่างไร

การย้ายคีย์เฟรมหลาย ๆคีย์เฟรมพร้อม ๆกัน

Moving Keys



1. บน Timeline นำ Cursor ไปวางไว้บนเฟรมแรกที่เราต้องการย้ายที่

2. กด Shift ค้างไว้ แล้วใช้เม้าส์ปุ่มซ้ายลากเป็นแถบ Highlight สีแดงขึ้นมาไปยังเฟรมสุดท้ายที่ต้องการเคลื่อนย้าย

3. สังเกตที่บริเวณกึ่งกลางของแถบ Highlight นี้จะมีหัวสามเหลี่ยมสีดำสองอันอยู่ชี้สวนทางกัน ให้เอาเม้าส์คลิกค้างตรงนี้ แล้วลากไปทางขวา สังเกตตัวเลขสีขาวที่อยู่ซ้ายสุดของแถบ Highlight ให้เปลี่ยนเป็นเลข 5 หมายถึงเฟรมที่ 5 จะเป็นเฟรมแรกของชุดคีย์เฟรมนี้ จากนั้นปล่อยเม้าส์ได้

4. จากตรงนี้จะพบว่า Timeline ของเราแสดงแค่ 24 เฟรม ทำให้มีเฟรมที่ตกขอบขวาไป 5 เฟรม เราจึงต้องขยายตัว Range Slider เพื่อให้แสดงผลครบทั้ง 28 เฟรม ทำได้โดยคลิกที่บริเวณสี่เหลี่ยมด้านขวาของ Range Slider แล้วลากไปทางขวาจนกว่าจะแดงตัวเลข 28 เป็นอันเสร็จสิ้นขั้นตอน

หมายเหตุ: ในการเคลื่อนย้ายคีย์เฟรมนั้น สามารถทำพร้อมๆกันทีละหลายๆคีย์เฟรม หรือจะย้ายทีจะคีย์เฟรมก็ได้ วิธีย้ายทีละเฟรมคือ ให้กด Shift ค้างไว้ จากนั้น Select คีย์เฟรมที่ต้องการ จะปรากฏแถบ Highlight สีแดงครอบเฉพาะคีย์เฟรมที่เลือก จากนั้นสามารถทำตามขั้นตอนที่ 3 เพื่อย้ายไปยังตำแหน่งที่ต้องการ

การยืดหรือหดคีย์เฟรมหลาย ๆคีย์เฟรมพร้อม ๆกัน

Rescaling Keys

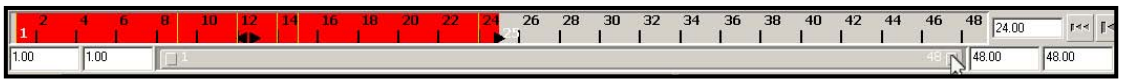
การย้ายคีย์เฟรมหลายๆอันพร้อมกัน สามารถทำได้เพื่อเปลี่ยนจุดเริ่มของเหตุการณ์ชุดนั้นๆ โดยที่ไม่มีผลต่อระยะเวลาของเหตุการณ์นั้นแต่อย่างใด (เหตุการณ์นั้นยังมีจำนวนคีย์เฟรมในชุดเท่าเดิม) ในการที่จะทำให้เหตุการณ์ในชุดนั้นสั้นหรือยาวขึ้นต้องใช้วิธี Scaling Keys

การ Scaling Keys คือการยืดหรือหดชุดของคีย์เฟรมใดๆในอัตราส่วนที่เท่ากัน มีผลคือทำให้ Motion ของแอนนิเมชันชุดนั้นเร็วขึ้นหรือช้าลงนั่นเอง เพื่อความเข้าใจเราจะลองทำดูจากแอนนิเมชัน Bouncing Ball ในหัวข้อที่ผ่านมา

*ถ้าใครเลื่อนคีย์เฟรมไป 5 เฟรม ให้เลื่อนกลับมายังจุดเริ่มต้นก่อน

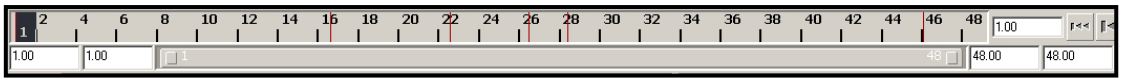
1. บน Timeline นำ Cursor ไปวางไว้บนเฟรมแรกของชุดแอนนิเมชันที่เราต้องหาร Rescale

2. กด Shift ค้างไว้ แล้วใช้เม้าส์ปุ่มซ้ายลากเป็นแถบ Highlight สีแดงขึ้นมาไปยังเฟรมสุดท้ายของชุด



3. ทำการเพิ่มความยาวของ Timeline เพื่อให้มีพื้นที่เพียงพอสำหรับการขยายคีย์เฟรม โดยคลิกที่บริเวณสี่เหลี่ยมด้านขวาของ Range Slider แล้วลากไปทางขวาจนกว่าตัวเลขเฟรมสุดท้ายจะเปลี่ยนเป็น 48

4. ที่บริเวณด้านขวาสุดของแถบ Highlight สีแดง จะมีหัวลูกศรสามเหลี่ยมชี้ไปทางขวา คลิกเม้าส์ที่สามเหลี่ยมนี้แล้วลากไปทางขวามือให้คลุมไปถึงเฟรมสุดท้ายคือเฟรมที่ 48



5. เมื่อเสร็จแล้วให้คลิกที่บริเวณใดๆบน Timeline แถบ Highlight จะหายไป สังเกตคีย์เฟรมทั้ง 6 คีย์ จะถูกยืดออกในอัตราส่วนที่เท่ากัน เมื่อเราลอง Preview ดูจะพบว่าลูกบอลลอยช้าลงเป็นสองเท่าจากเดิม (เนื่องจากตอนแรกใช้ 24 เฟรม แต่ตอนนี้แอนนิเมชันเรามี 48 เฟรม)

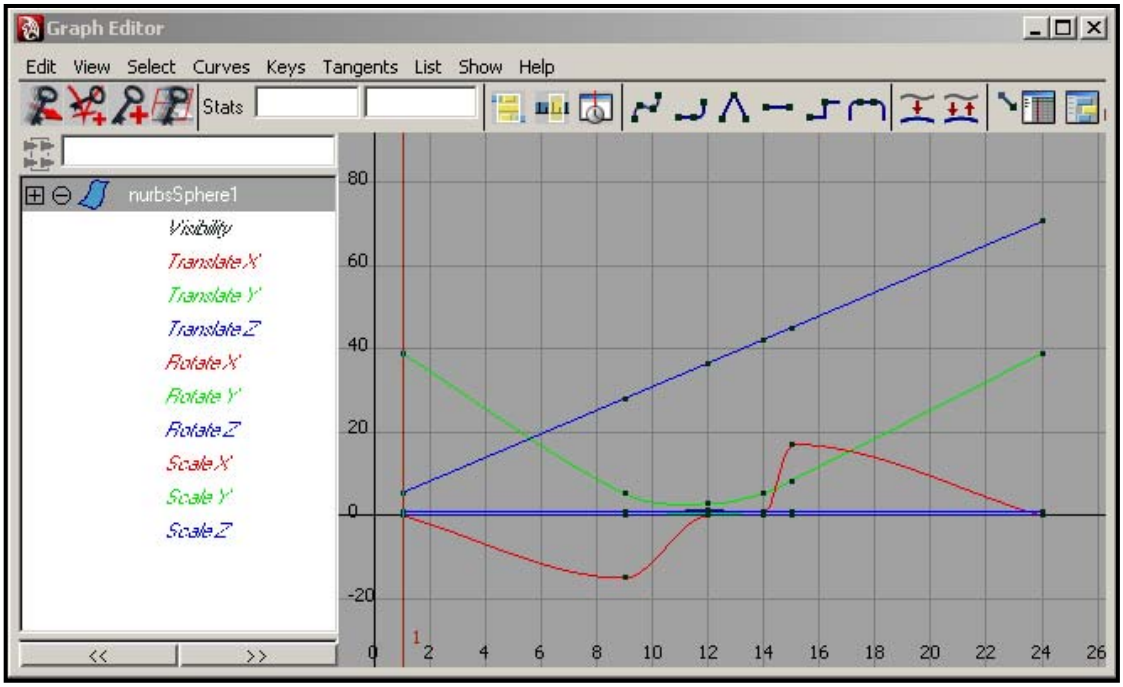
การแก้ไขการเคลื่อนไหวของวัตถุด้วยกราฟ

Graph Editor for Animation

อย่างที่ทราบกันว่าการสร้างแอนนิเมชันเราสามารถทำได้โดยการใส่คีย์เฟรม เพื่อกำหนดการเคลื่อนไหวหลักๆ ของวัตถุ ในขณะที่โปรแกรมจะคำนวณค่า In-Between ให้กับเรา ซึ่งก็ดูสมเหตุสมผลดี แต่ปัญหาอยู่ที่แล้วเราจะมั่นใจได้อย่างไรว่าค่า In-Between เหล่านั้นถูกต้องตรงความต้องการของเรา ยกตัวอย่างเช่นการสร้างแอนนิเมชันของขบวนรถไฟ แล่นจากจุด A ไปยังจุด B เราสามารถทำได้โดยการใส่คีย์เฟรมที่จุด A และ B แล้วปล่อยให้รถไฟแล่นไปตามค่า In-Between ที่โปรแกรมคำนวณให้ ผลลัพธ์ที่ได้คือขบวนรถไฟที่แล่นด้วยความเร็วคงที่ตั้งแต่เริ่มออกตัวไปจนจอด แล้วมันมีอะไรไม่ถูกต้องหรือ ในความเป็นจริงนั้นในจังหวะที่รถไฟกำลังออกตัวจะมีความหน่วงอยู่ ทำให้ความเร็วช้า พอแล่นไปสักพักความเร็วจะเพิ่มขึ้น และจะชะลออีกครั้งก่อนที่จะจอดเทียบชานชลา จึงเป็นไปไม่ได้ที่จะแล่นด้วยความเร็วคงที่ตั้งแต่ต้นจนจบ

ปัญหาเหล่านี้สามารถแก้ไขได้ด้วย Graph Editor โดยที่โปรแกรมจะทำการคำนวณค่าการเคลื่อนไหวที่ไม่มีคีย์เฟรมเหล่านี้ (In-Between) ออกมาในรูปแบบของกราฟเส้น เราสามารถเข้าไปทำการแก้ไขสร้างค่า Slow-In/ Slow-Out ต่างๆได้ตามต้องการ

เราสามารถเปิดหน้าต่าง Graph Editor นี้ได้โดยการไปที่ **Window/ Animation Editors/ Graph Editor** สังเกตจากภาพตัวอย่างที่ 7.6 หน้าตาของหน้าต่างแก้ไขกราฟนี้สามารถแบ่งเป็นสามส่วนหลักๆ ส่วนแรกคือส่วนของชุดคำสั่งในการ Edit และไอคอนลัดต่างๆทางด้านบน ส่วนที่สองคือหน้าต่างสีขาวทางซ้ายมือ แสดงประเภทของการเคลื่อนไหวแบบต่างๆแบ่งออกเป็นสีตามแกน เช่น การเคลื่อนที่, การหมุน, การเปลี่ยนขนาดในแนวแกน X เป็นสีแดง ส่วนสีเขียวและน้ำเงินจะแสดง การเคลื่อนที่, การหมุน, การเปลี่ยนขนาดในแนวแกน Yและ Z ตามลำดับ และในส่วนที่สามคือหน้าต่างสีเทาขวามือ เป็นพื้นที่แสดงลักษณะของเส้นกราฟ โดยจะมีสีของกราฟเช่นเดียวกับสีของแกน เพื่อให้ผู้ใช้สามารถจำแนกได้อย่างไม่สับสน



ภาพประกอบที่ 7.6 แสดงหน้าต่าง Graph Editor

ประเภทของเส้นกราฟ

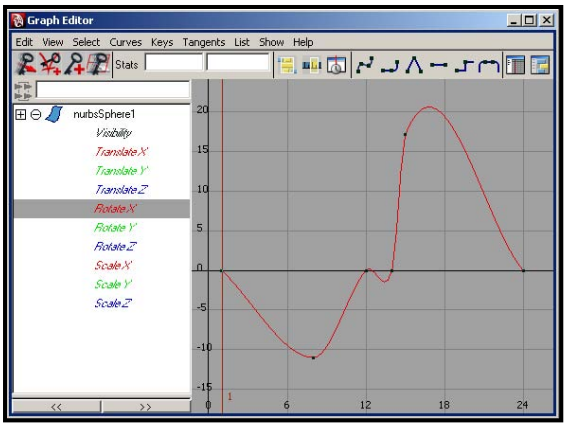
Tangent Types

เส้นกราฟมีหลายแบบ แบ่งตามลักษณะความโค้ง ซึ่งลักษณะความโค้งเหล่านี้สามารถปรับได้จาก Tangent Icons บริเวณด้านบนของหน้าต่าง Graph Editor สังเกตภาพประกอบที่ 7.7 ซึ่งแต่ละตัวจะให้คุณสมบัติของเส้นโค้งแตกต่างกัน การปรับความโค้งนี้มีผลโดยตรงกับการเคลื่อนที่ของวัตถุ ว่าจะมีความ Smooth หรือแข็งอย่างไร

ภาพประกอบที่ 7.7 แสดง Tangent Icons

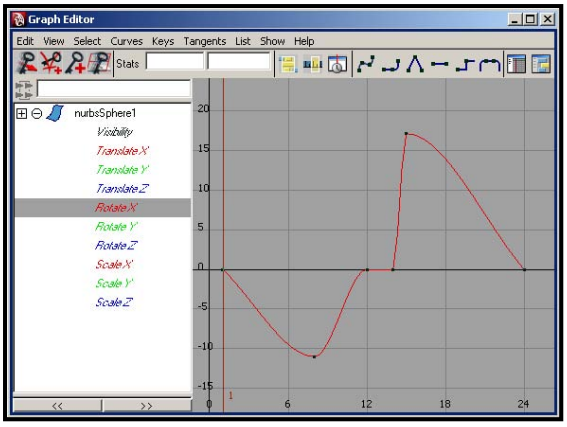
เราสามารถเปลี่ยนลักษณะของเส้นกราฟได้ โดยการ Drag Mouse ครอบเส้นกราฟในส่วนที่เราต้องการเปลี่ยนความโค้ง เส้นกราฟที่ถูกเลือกจะเปลี่ยนเป็นสีขาว จากนั้นก็คลิกที่ไอคอน Tangent ที่ต้องการ กราฟจะถูกเปลี่ยนลักษณะความโค้งไปเป็นแบบที่เลือก แล้ว Tangents แต่ละตัวมีคุณลักษณะอย่างไรบ้างล่ะ เราจะมาดูกัน

Spline Tangents



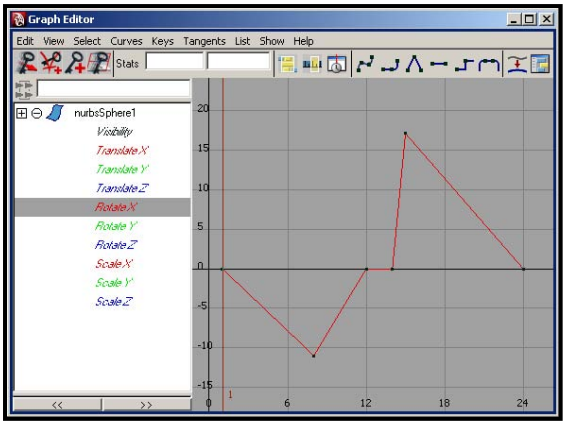
Spline Tangents สร้างเส้นกราฟที่มีความ Smooth ต่อเนื่องกัน ทำให้การเคลื่อนไหวของวัตถุมีความกลมกลืน เหมาะกับการสร้างการเคลื่อนไหว ของสิ่งที่เคลื่อนที่อย่างต่อเนื่องอยู่ตลอดเวลา เช่น ปลาที่ว่ายอยู่ในน้ำ ใบไม้ที่ร่วงลงสู่พื้น

Clamped Tangents



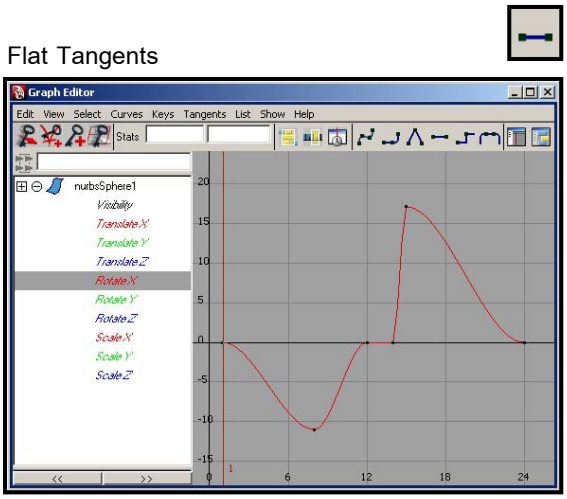
Clamped Tangents เป็นลักษณะกราฟตั้งต้นของโปรแกรม มีลักษณะคล้ายกับ Spline Tangents มากแตกต่างกันแค่ ถ้ามีการคงที่ของวัตถุที่ตำแหน่งใดตำแหน่งหนึ่ง Clamped จะให้เส้นกราฟเป็นเส้นตรงได้ นั่นหมายความว่าวัตถุจะหยุดนิ่งไม่มีการเคลื่อนไหว (สังเกตเฟรมที่ 12 -14 จากภาพตัวอย่าง) ในขณะที่ Spline จะสร้างเป็นเส้นโค้ง หมายความว่าใน Spline วัตถุจะไม่สามารถหยุดแบบแข็งที่อได้เลย

Linear Tangents



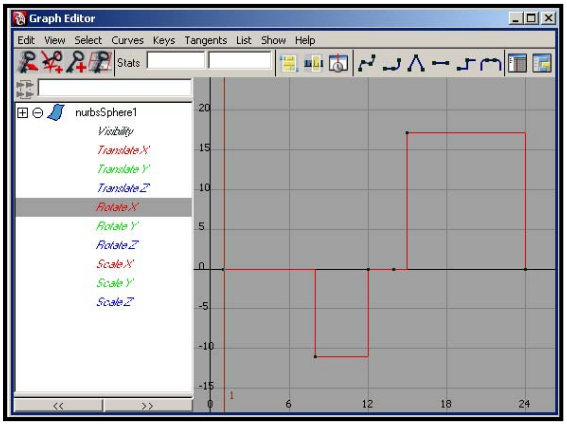
Linear Tangents ให้เส้นกราฟที่มีลักษณะเส้นตรง มีการหักมุมเป็นมุมแหลม โดยไม่มีส่วนโค้งแต่อย่างใด เหมาะสำหรับการเคลื่อนที่ของเครื่องจักร หุ่นยนต์ หรือวัตถุ ที่ต้องการความเร็วตกกระทบเท่ากับความเร็วสะท้อน เช่นการกระแทกกันระหว่างลูกบอลเหล็ก กับผนังเหล็ก หรือจังหวะที่ลูกปิงปองกระทบหน้าไม้ปิงปองเป็นต้น

Flat Tangents



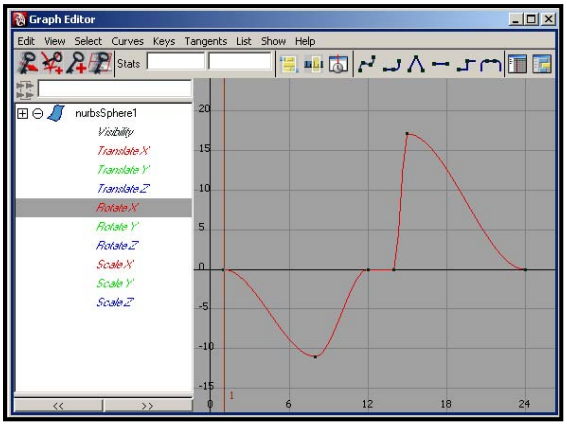
Flat Tangents จะสร้างเส้นกราฟที่มีลักษณะคล้ายกับ Clamped แตกต่างที่ Flat จะสร้าง Slow-in, Slow-out ให้กับวัตถุ หมายความว่าในระหว่างสองคีย์เฟรม วัตถุจะเริ่มเคลื่อนที่ช้า ก่อนจะเพิ่มความเร็วขึ้นในช่วงกลาง แล้วจะลดความเร็วลงอีกครั้งก่อนที่จะผ่านไปยังคีย์เฟรมถัดไป

Step Tangents



Step Tangents จะสร้างกราฟเส้นตรงในมุม 90 และ 180 องศาเท่านั้น ทำให้กราฟมีลักษณะเหมือนขั้นบันได กราฟลักษณะนี้จะทำการแช่แข็งวัตถุไว้ที่ตำแหน่งหนึ่ง จนกว่าจะผ่านไปถึงคีย์เฟรมถัดไป (วัตถุจะหยุดนิ่ง แล้วพอถึงคีย์เฟรมถัดไป ก็จะกระโดดไปโผล่ที่ตำแหน่งใหม่เลย) เหมาะกับการใส่คีย์เฟรมให้กับกล้อง (Camera) ที่ต้องการหยุดกล้องเพื่อถ่ายฉากใดๆไว้เวลาหนึ่ง ก่อนที่จะกระโดดไปถ่ายมุมอื่นในคีย์เฟรมถัดไปทันที

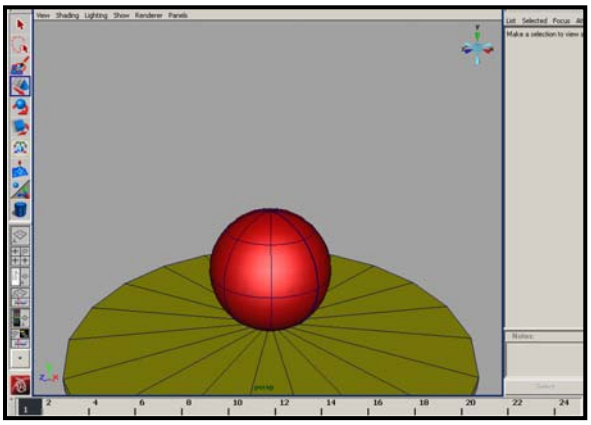
Plateau Tangents



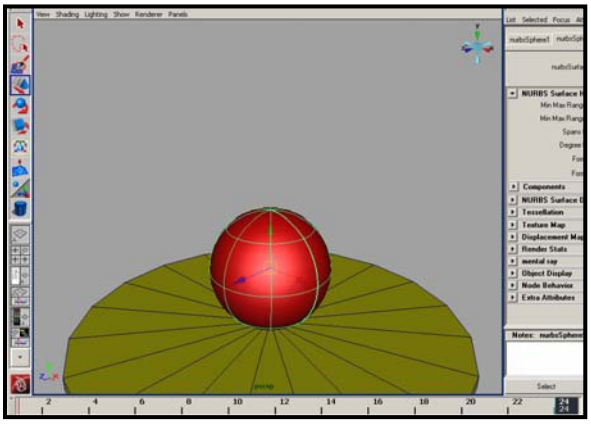
Plateau Tangents มีลักษณะคล้าย Clamped แตกต่างกันตรงที่ Plateau จะสามารถบังคับไม่ให้วัตถุเลยออกจากตำแหน่งที่ต้องการได้ ยกตัวอย่างเช่นการสร้างลูกบอลที่เด้งกระทบพื้นแล้วกระดอนขึ้นมา Clamped จะสร้างเส้นโค้งขึ้นมาในจุดที่ลูกบอลกระแทกพื้น แล้วกระเด็นขึ้นมา จากลักษณะของเส้นกราฟ ในสองคีย์เฟรมนี้มีโอกาสเป็นไปได้ที่ลูกบอลจะเลยเข้าไปในผนังนิดนึง การใช้ Plateau จะช่วยป้องกันเหตุการณ์แบบนี้ได้

ทดลองแก้ไขกราฟการกระเด้งของลูกบอล

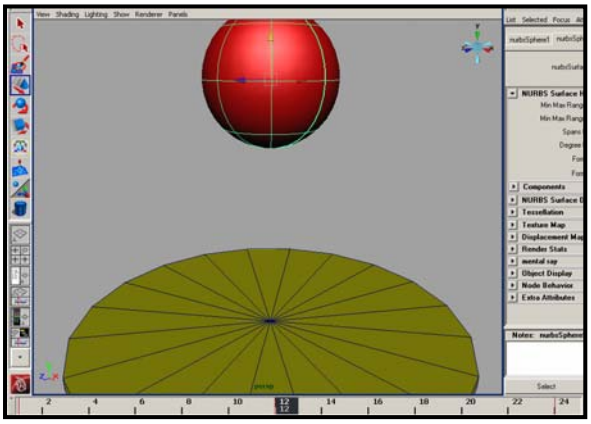
Graph Editor for Bouncing Ball



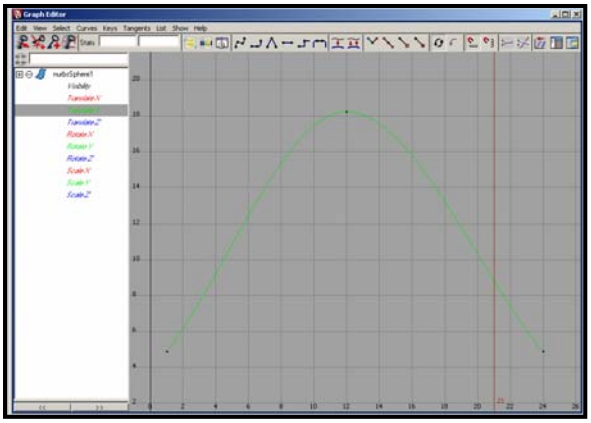
1. สร้างฉากที่ประกอบด้วยลูกบอล และพื้นสำหรับให้ลูกบอลเด้งขึ้นมา ดังภาพตัวอย่าง เสร็จแล้วให้เลื่อนหัวอ่านไปที่เฟรมที่ 1 ทำการ Select ที่ลูกบอล แล้วกด “S” เพื่อใส่คีย์เฟรม



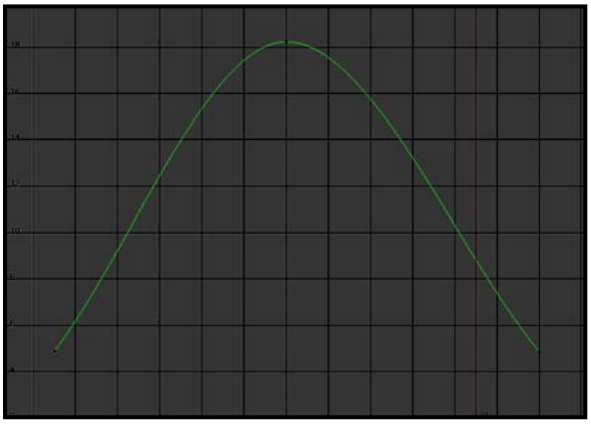
2. เลื่อนหัวอ่านไปยังเฟรมที่ 24 แล้วทำการใส่คีย์เฟรมให้กับลูกบอลอีกทีหนึ่ง



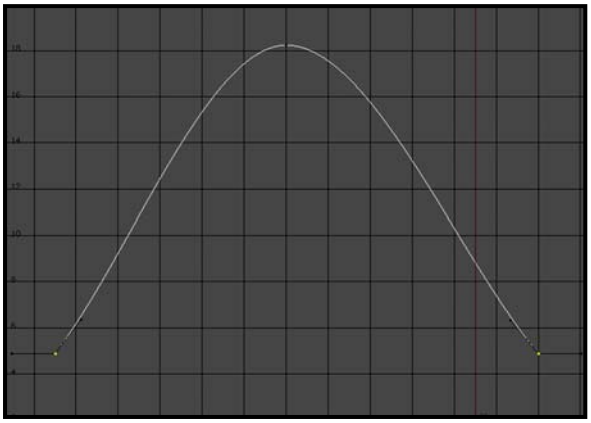
3. เลื่อนหัวอ่านกลับมายังเฟรมที่ 12 ใช้ Move Tool ยกลูกบอลขึ้นดังภาพตัวอย่าง จากนั้นทำการใส่คีย์เฟรมให้กับลูกบอล ลอง Preview ดู จะได้ลูกบอลตกกระทบพื้น แล้วเด้งขึ้นมาที่ตำแหน่งเดิม จากคีย์เฟรมทั้งสามที่เราใส่เข้าไป



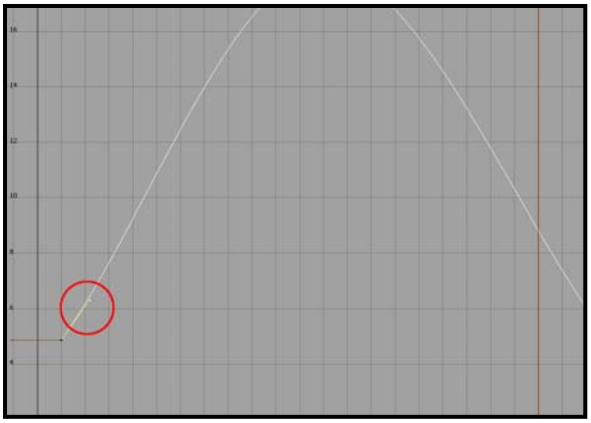
4. ไปที่ **Window/ Animation Editors/ Graph Editor** เพื่อเปิดหน้าต่าง Graph Editor ขึ้นมา จะพบว่า มีเพียงแค่เส้นกราฟสีเขียวอันเดียวที่ไม่เป็นเส้นตรง ทั้งนี้เนื่องจากวัตถุมีการเคลื่อนที่ในแนวแกนเดียว นั่นคือแกน Y (เคลื่อนที่ขึ้นแล้วก็ลง) ให้เลือกไปที่หัวข้อ Translate Y (สีเขียว) จากหน้าต่างสีขาวด้านขวามือ เพื่อให้หน้าจอแสดงผลของกราฟนี้เท่านั้น



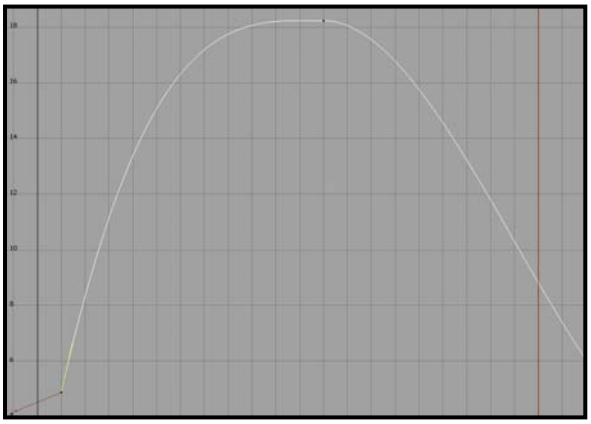
5. กดปุ่ม “F” บนคีย์บอร์ด เพื่อทำการ Frame กราฟเส้นนี้ ให้แสดงผลเต็มหน้าจอแสดงผล เพื่อความสะดวกในการแก้ไข สังเกตจากเส้นกราฟที่มีลักษณะโค้งเล็กน้อยบริเวณตรงกลาง แสดงว่าลูกบอลจะลอยขึ้นมาด้วยความเร็วคงที่ และจะมีการชะลอตัวเล็กน้อยก่อนที่จะตกลง ลองกด Preview ดู สังเกตความสัมพันธ์ระหว่างลูกบอลกับเส้นกราฟ



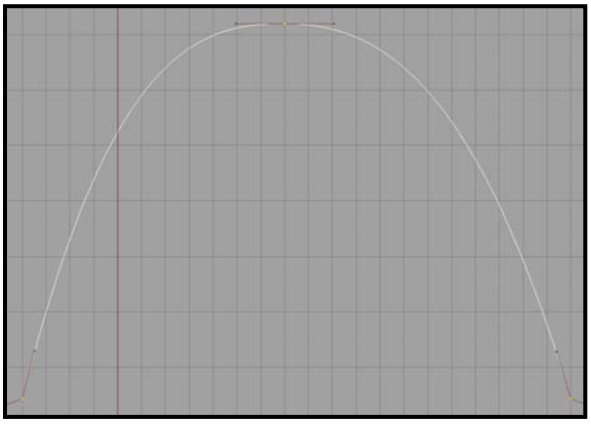
6. ในความเป็นจริงเมื่อลูกบอลตกกระทบพื้นจะกระดอนออกไปอย่างรวจเร็ว มีลักษณะเป็นมุมแหลม ดังนั้นให้ Drag Mouse ครอบจุดเริ่มต้นและจุดสุดท้ายของกราฟ (เฟรมที่ 1 กับ 24) จุดที่เลือกจะเปลี่ยนเป็นสีเหลือง ในขณะที่เส้นกราฟเปลี่ยนเป็นสีขาว จากนั้นให้กดไปที่ไอคอน Linear Tangents ขาของตัวปรับกราฟจะหักเป็นมุมแหลม



7. Drag Mouse ครอบจุดแรกของกราฟ เพื่อทำการ Select เฉพาะจุดนี้ จะปรากฏขาสีเลือดหมูของข้างออกมา ให้ Drag Mouse ครอบขาที่อยู่ขวามือ ขาข้างนั้นจะกลายเป็นสีเหลือง การปรับแต่งเส้นกราฟสามารถทำได้โดยการปรับขาทั้งสองข้างของแต่ละจุดเหล่านี้



8. ไปที่หน้าจอปกติของโปรแกรม เลือกไปที่ Move Tool จากนั้นกลับมาที่หน้าต่าง Graph Editor ใช้เม้าส์ปุ่มกลางลากปรับแต่งขาข้างที่ถูกเลือกอยู่ ให้เส้นกราฟมีลักษณะดังภาพตัวอย่าง



9. จากนั้นทำการปรับจุดสุดท้าย (เฟรมที่ 24) เช่นเดียวกับจุดแรก ปรับแต่งให้ได้ลักษณะของกราฟดังภาพตัวอย่าง จากนั้นลอง Preview ดู จะพบว่าลูกบอลกระเด้งขึ้นลงที่กระชับขึ้น ในขณะที่มีการชะลอตัวค้างก่อนที่จะตกลงมาเพิ่มขึ้น ทั้งนี้เนื่องจากเส้นกราฟที่มีความชันเพิ่มขึ้นในส่วนหัวกับท้าย ในขณะที่มีความชันน้อยลงบริเวณกึ่งกลาง (เวลาที่ลูกบอลลอยถึงจุดสูงสุด)

จากการ Edit Graph แบบนี้ จะให้ความรู้สึกของลูกบอลยางที่มีความยืดหยุ่นตกกระทบพื้นมากขึ้น แน่นอนว่ากราฟลักษณะนี้ไม่เหมาะกับวัตถุที่มีความแข็ง หรือมีความยืดหยุ่นน้อย ซึ่งกราฟลักษณะ Linear Tangents น่าจะมีความเหมาะสมกว่า ทั้งนี้ผู้ใช้จะต้องตัดสินใจโดยคำนึงถึงคุณลักษณะของวัตถุและพื้นผิวให้ดี เพื่อให้การแก้ไขกราฟเป็นไปอย่างสมจริง และให้ผลลัพธ์ตรงตามความต้องการ