Dynamics in Maya

Gary Monheit

Alias / Wavefront
Dynamics in Maya

- Overall Requirements
- Architecture and Features
- Animations
Overall Requirements

- Why Dynamics?
- Problems with traditional animation techniques
- Kinds of dynamics problems
- Requirements for software architecture
Why Dynamics?
Problems with traditional animation

- Hard to keyframe
- Rendered polygons don’t give desired effects
- Amorphous effects hard to achieve
- Lacks high-level control, simulation
- Lacks procedural control
Kinds of dynamics problems

Particles

- Contain a set of points in global space with attributes
- Affected by dynamic solver
- Created at either object creation time or by emitters
- Handle birth and death
- Support a variety of rendering techniques

Soft Bodies

- Geometry with points updated by a particle object
- Superset of all particle features

Rigid Bodies

- Geometry with fixed topology and points
- Affected by collisions, constraints, and external forces
Requirements for software architecture

- Integrated
- Extensible
- Reusable
- Tweakable
- Support Relationships
Integrated

Maya as scene object architecture
  • Nodes
  • Attributes
  • Connections

Keyframing
  • Any scalar attributes
  • 3D motion of non-dynamic objects
  • Keyframe editors are familiar to animators

Layered procedural animation
  • Output from *dynamics* can be input to other nodes
  • Outputs from other nodes can be input to *dynamics*

Rendering
  • Rendered within Maya or exported to other renderer
  • Software rendering (volumetric)
  • Hardware rendering (openGL)
Extensible

- Maya embedded language (Mel)
- Parameter driven: attributes
- ASCII scene file format
Reusable

- Layered scenes with geometry and effects
- Create your own clipFX with Mel
Tweakable

Attributes are modifiable

- Keyframes
- Expressions
- Connect to other nodes
- Add user-defined attributes
Relationships

- Object-to-object relationships
- Attribute-to-attribute relationships
Dynamics in Maya

- Overall Requirements
- Architecture and Features
- Animations
Architecture & Features

- Maya Architecture
- Dynamic Architecture
- Rendering
- Expressions
- Integrated Dynamics
Maya Architecture (part 1)

- Nodes
- Attributes
- Connections
Maya Architecture (part 2)

- Transforms and Shapes
  - DAG: Directed Acyclic Graph
  - The "object" is really the transform plus the shape(s)
Dynamics Architecture

- Particles
- Fields
- Emitters
- Collision Models
- Particle Collision Events
- Controllers
- Connectables and Connections
- Soft Bodies
- Springs
- Rigid Bodies
Particles

- A particle object is a particle system
- Per-particle (array) attributes vs. scalar attributes
- User-defined attributes
- Particles act independently
- No particle-to-particle collisions
- What affects particles?
  - Fields
  - Springs
  - Attribute values
  - Expressions (one-line or multi-line)
  - Dynamics controller attributes
Fields

- Positional fields
- Geometry fields. Example: sphere owns radial field
- Types of fields
  - Air
  - Drag
  - Gravity
  - Newton
  - Radial
  - Turbulence
  - Uniform
  - Vortex
Emitters

Emitters
• Creates new particles into a particle object
• Hose vs. water. Emitter is the hose.

Positional emitter
• The transform defines the position, orientation
• Parented, keyframed, in IK chain, ...

Geometry emitter
• Geometry owns the emitter
• Point vs. surface
• Particle, NURBS, polygon, lattice, curve ...
Collision Models

- Elasticity
- Friction
- Collision trace depth
Particle Collision Events

- In the event that a particle collides:
  - Emit
  - Split
  - Die

- Optionally call a Mel proc each time the event occurs
Connectables and Connections

Dynamics object \[\rightarrow\] Connectable object

Dynamics objects:
- Particles
- Soft bodies
- Rigid bodies

Connectable object owns one or more of any combination of:
- Fields
- Emitters
- Collision models

What kind of object can be a connectable:
- Polygon, NURBS, lattice, curve, ...
### Examples of connections

<table>
<thead>
<tr>
<th>Description</th>
<th>Conn Type</th>
<th>Connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere emits particles</td>
<td>emission</td>
<td>particles → sphere</td>
</tr>
<tr>
<td>Particles fall under gravity</td>
<td>field</td>
<td>particles → gravity</td>
</tr>
<tr>
<td>Particles bounce on surface</td>
<td>collision</td>
<td>particles → surface</td>
</tr>
<tr>
<td>Particles owning radial field repel a soft cube</td>
<td>field</td>
<td>soft cube → particles</td>
</tr>
<tr>
<td>Point emitter emits soft mesh</td>
<td>emission</td>
<td>soft mesh → pt emitter</td>
</tr>
</tbody>
</table>
Controllers

- Manage dynamic objects that may interact
- Interface to a differential equations solver
- Necessary so that all dynamics objects get updated in sync
- Hold global attributes for dynamics
Goals

- Goals are geometry, particles, or soft bodies referenced in a particle object
- Create a force for each particle to move toward a point on the goal
- Particles or soft bodies can have multiple goals
- Does not require the same number of points as target
- Individual goal weight 0 to 1
- Attribute goalPP 0 to 1 (per-particle)
- Dynamics weight 0 to 1 for the forces generated by fields and collisions
Springs

- Classical mechanics spring
- Hook’s Law
- Stiffness and damping
- Start force weight, end force weight
  - 0 to 1
  - Amount of the force of the spring that gets applied to either start or end point
  - (0, 1) or (1, 0) simulates nailing one end of the spring
Soft bodies

- *Soft* is a command, not a node
- For the user, *soft bodies* are geometry whose points are updated by a particle object
- Internally, *soft bodies* are the particle object
- Geometry converted or duplicated in a *soft body*
- Springs generated by min–max distance or by walk length
- Oversampling (small solver step size) stabilize springs
Rigid Bodies

- Nodes
  - Rigid body
- Constraints
  - Nail
  - Pin
  - Hinge
  - Spring
  - Barrier
- Transformations updated by a solver that applies forces
- Internal constraints added implicitly (by contact) or explicitly (by constraint objects)
Rigid Body Features

- Contact forces handled for resting and sliding objects
- Static and dynamic friction
- Active rigid body
  - collision detection
  - dynamic response
- Passive rigid body
  - collision detection
  - independent control by keyframes, expressions, other nodes
- Cache motion
- Collision layers used to reduce complexity
- Choice of 3 kinds of solvers
Rendering

- Render Types
- Hardware Rendering in OpenGL
- Software Rendering
Render Types

- Point
- Multi-point
- Streak (lines)
- Multi-streak
- Sprites (textures)
- Sphere
- Geometry
- Numeric (for debugging)
- Tube (Software)
- Cloud (Software)
- Blobby Surface (Software)
Software rendering

- Volumetric rendering
- Blobby surface render type uses any shader
- Tube and Cloud render types use specific particle cloud shader
- Future: hair shader
Expressions

- Mel Language
- Attribute Expressions
- Particle Expressions
Mel Language

- Maya Embedded Language
- Unified language for file format, command engine, and expressions
- Like C
- Like shell
Attribute Expressions

- Lazy evaluation – evaluate when needed
- Multi-line expressions
- Multiple input and output attributes
Particle Expressions

- Particle expressions executed per-particle
- Multi-line expressions are like small programs
- Any attribute accessible for setting and getting values
- Run-time vs. creation rules
- Built-in functions for math, system, I/O, arrays
- Data types: int, float, string, vector
  - int[], float[], string[], vector[]
- Maya commands can be called from expressions

Example:

```
// Creation rule for setting color:
dynExpression -c -at rgbPP -s "rgbPP = sphrand(1);"
  colorCloud;

// Runtime rule for setting color:
dynExpression -r -at rgbPP -s
  "if( age > .5 ) rgbPP += <<0,0,rand(.1)>>;"
  colorCloud;
```
Integrated Dynamics

• Dynamics and transforms
• Rigid Bodies – active and passive
• Soft Body, curves, and IK chain
• Soft Body and deformers
• Soft Body, Rigid Body, and goals
• Particle collision events ...
Dynamics in Maya

- Overall Requirements
- Architecture and Features
- Animations
Future Work

- Maya 2.0