Week 07  MEL for particles and dynamics

- Expression for particles: introduction
- Per-particle attributes
- Different kinds of particle expression
- List of particle attributes

Expression for particles: introduction

Application examples

It's quite typical to use expressions to control particles. For example:
- Create sophisticated particle motion and life span; for example, the movement and fading of exploding fireworks.
- Create complex colors for emitted particles, for example, rocket exhaust flames.
- Create complex colors and movement of particles following collision with geometry, for example, sparks resulting from fired bullets.

Create particle in Maya

You can have at least 2 ways to create particles through the Maya interface:

1. Particles > Particle Tool
   - Then click on the view window at different positions, and press ENTER. This will create a particle node containing several particles.

2. Particles > Create Emitter
   - This will create an emitter node and a particle node. The emitter node controls how many particles should be emitted from the particle node.

In all the exercises today, we will simply use the second method – i.e. create emitter.
More about create emitter

You can fine-tune the options before you create the emitter. For example, you may try to have **Emitter Type:Volume** and change the **Volume Shape:Cylinder** to see what happen. Also try to scale the green cylinder created to see what happen.

**About Field**

You can create “field” to affect the particles motion. Let’s try a simple example:

Step 1) Create a emitter by using **Particles > Create Emitter**.
Step 2) Select the node **particle1**, and select **Fields > Air > option**.
Step 3) Set the DirectionY to 0, DirectionZ to -1. Play the scene to see what happen.
*Maya particles* will be covered in your animation workshops, so I will stop here, and not going to cover the details in this course. I will only focusing on how to use expression to control the particle effects.

**More about Maya node’s attribute**

If you don’t know the name of a node’s attribute, you can right-click on the name of the attribute and choose “Create New Expression…”. Then Maya will open the expression editor with the name of the attribute shown in the editor.

Of course, we can write expression to control any attributes of the emitter node and the particles node, in the same way as what we did in the last week. However, there are other issues for writing particle expression, which will be covered in today’s class.
Before we look at the other issues, let’s try a simple exercise first.

**Class exercise: emitter’s attributes**

**Exercise 1**
Step 1) Create a emitter by using **Particles > Create Emitter**
Step 2) Set the **Speed** attribute of the emitter to 10.
Step 3) The **Rate** attribute of an emitter controlling how many particles should be emitted per second. Write an expression for the emitter’s **Rate** attribute:

```java
if ( (frame % 20) == 0 ) {
    emitter1.rate = 300;
} else {
    emitter1.rate = 0;
}
```

(Run the scene to see what will happen. You may need to use a longer time range.)

**Exercise 2**
Using similar idea, try to create the effect (i.e. particles emitted when the ball hits the ground) as shown in the demo file J:\SM3122\Week07-dust.avi.

(Hints: in this simple exercise, you are **not** required to use collision. Just use the ball’s y-position to decide whether the emitter should emit particle or not.)
Per-particle attributes

In the previous exercises we modify the emitter’s attribute, not the particle’s attribute. Now, click on the particle1 node, and open the Attribute Editor. Choose the particleShape1 tab. You can find that particle also has a lot of attributes.

But don’t forget that each particleShape node should contain many particles. So, which objects are those “attributes” belongs to? Belongs to the particleShape? Or belongs to each individual particle?

Particle’s attributes are divided into two categories:

- **Per-object attribute**: each particleShape has attributes that are shared by all particles in that particleShape.
- **Per-particle attribute**: each particle (i.e. the tiny points) in the particleShape has its own value. For example, each particle has its own position and velocity.

When you want to set values or write expression for particle attributes, make sure that you understand whether they are per-object or per-particle attributes.
Note:
- You can control per-object attributes through the Maya interface.
- Though per-particle attributes are best for creating complex effects, you can't set their value through Maya interface. You can only use expression to control per-particle attributes.
- If both exist, the per-particle attribute takes precedence. For example, the per-particle color overrides the per-object color.
- Most (but not all) per-particle attribute has PP at the end of the attribute name.

Some of per-particle attributes are available by default, but some of them are “hidden”. They can only be available when you use the “Add Dynamic Attributes” to “add” it:

Click this button  Choose the “hidden” attribute to add
Now, let’s do a simple exercise on writing expression for per-particle attribute.

**Exercise 1**

1. Create a emitter by using **Particles > Create Emitter**
2. Select the node **particle1**, and in the Attribute Editor, choose the **particleShape1** tab, and find the **position** under **Per Particle (Array) Attributes**.
3. Right-click at the box besides **position**, and choose **Runtime Expression Before Dynamics** (I will explain what is “runtime expression before dynamics” soon).
4. Type (and create) the following expression in the Expression Editor:

```
vector $oldPos = particleShape1.position;
float $newY = ($oldPos.x)*($oldPos.x) + ($oldPos.z)*($oldPos.z);
particleShape1.position = <<- $oldPos.x, $newY, $oldPos.z >>;
```

(Run the scene to see the result. Make sure that you have a long enough timeline range.)

Note that in this exercise, the **particleShape1.position** is a per-particle attribute.

**Exercise 2**

Using similar idea, try to create the effect (i.e. particles going out in a sin-wave manner) as shown in the demo file **J:\SM3122\Week7-particle-wave.avi**.

(Hints: you can try $newY = sin( XXX ). What should be “XXX”?)

**Exercise 3**

Try to create the same effect as shown in the demo file **J:\SM3122\Week7-exercise.avi**.

**Hint #1**

To move a point in a circular motion, the velocity at position (x, y) should be:

```
particleShape1.velocity = <<- -1*speed*y , speed*x, 0 >>;
```

where speed is a variable to control the speed of the motion. The larger speed, the circle will become larger and larger. Note that the speed can also be negative.

**Hint #2**

In the particle expression, divide the effect based on the current frame number, say:

```
if (frame < 90) {
    // create a sin() wave effect, similar to the previous exercise
} else {
    // create a circular motion by changing particle’s velocity,
    // with speed changing based on time
}
```
Different kinds of particle expression

Right-click at the box besides a per-particle attribute such as **position**, you will find **Creation Expression**, **Runtime Expression Before Dynamics**, and **Runtime Expression After Dynamics**. So, what’s the different between them?

Maya creates particle effects in the following flow:

1. **Creation Expression**: your expression will be executed **once**, when particle **starts** to come out (i.e. at position #1 above). It is usually used for initialization.

2. **Runtime Expression Before Dynamics**: your expression will be executed **every frame, before** Maya calculates the dynamic (i.e. at position #2 above). Executing before the dynamics engine gives you a chance to alter particle positions, velocities, or accelerations, while still allowing the dynamic engine to have other dynamics effect (say, gravity, field, or collision).

3. **Runtime Expression After Dynamics**: your expression will be executed **every frame, after** Maya calculates the dynamic (i.e. at position #3 above). Executing after the dynamics engine allows your expressions to completely override the dynamic engine's calculated particle motion.

For example, in **J:\SM3122\Week07-test.mb** I have a scene with an emitter, a particle node, and a gravity field (created by selecting **particle1** and choose the menu item **Fields > Gravity**). Try the following expression to see different effects:

```cpp
// Creation expression: randomize particles’ initial position
particleShape1.position = << rand(-5,5), 0, rand(-5,5) >>;
```
Make sure that you understand the different between these 3 kinds of expression.

**List of particle attributes**

The Maya online manual provides has a page showing all the particle attributes:
1. From the menu bar, choose Help > Maya Help
2. On the right side, locate:
   - Using Maya > General > MEL and Expressions > Particle expressions > Assign to vectors and vector arrays > List of particle attributes

**Remarks for position, velocity and acceleration attributes**

Amount all the per-particle attributes, in particular, we are usually interested in the particles’ attributes **position**, **velocity** and **acceleration**. Here are some useful remarks:

- To give a smooth, random motion to particles with a runtime expression, assign random numbers to the particle shape's acceleration attribute. A change in acceleration always gives smooth motion no matter how abruptly its value changes.
- To give a jittery random motion to particles with a runtime expression, assign random numbers to the particle shape's velocity or position attributes.
- If you want to give particles a constant acceleration, assign the acceleration attribute a constant value in a runtime expression rather than in a creation expression. The acceleration attribute works differently than the position or velocity attributes in an important way. **Maya initializes per-particle’s acceleration value to \(<<0,0,0>>\) before each frame.**
Particle goal

We mentioned that particle has a lot of attributes that are “hidden” by default, and they are available only after you “Add Attribute”, and you can control them only through Expression.

In this section we will look at an example on how to use one of the “hidden” particle attribute, called “particle goalU/goalV”. In the Maya GUI, “particle goal” is available, but has limited features. We can use Expression to extend the features.

First, let’s see what Maya “particle goal” provides, without using Expression:
1. Open the scene J:\SM3122\Week7-nurbsHead.mb.
2. Create an Emitter (Particles > Create Emitter).
3. Select the particle1, and shift-select one of the NURBS surface (say, surface).
4. Create particle goal (Particles > Goal > option).
5. Set the Goal Weight to be 1, and create the goal.

When goal weight is zero, the goal doesn’t have effect. When goal weight equals to 1, the “goal” will suppose “attract” the particles to the surface. However, by default it will attract particles to the CV of the NURBS, but not on the surface of that object.

To solve this problem, we can use the “hidden” attribute goalU and goalV. GoalU and GoalV specific which (U,V) location the particle should attach to the surface. (Each particle should have different goalU and goalV, so, it is a per-particle attribute.)

We can use an Expression to “randomize” the location.
1. First, add the per-particle attributes goalU and goalV using the technique we shown on page 6.
2. Second, create a “creation expression” and type the following, to see what happen:

\[
\begin{align*}
\text{particleShape1.goalU} &= \text{rand}(0,9); \\
\text{particleShape1.goalV} &= \text{rand}(0,19);
\end{align*}
\]

(Question: why 9 and 19?)
Class exercise: particle goal
(Need to hand in at the end of the class)

Given the scene file J:\SM3122\Week7-nurbsHead.mb, try to create the particle effect as shown in the demo file J:\SM3122\Week7-particle-goal.avi.

Hint:
1. You can create several emitter and particles objects.
2. For each particle object, assign one NURBS surface as its goal.
3. To have better effect, you can set each Emitter Type to “Volume: Cube”.
4. Add an “Air” Field to “blow away” the particles after frame 90. To achieve this, you can animate the “Goal Weight” (just by key-framing). Before frame 90, the goal weight is set to 1. After frame 90, the goal weight is set to zero.
5. Hide the original NURBS surfaces.

Save the finished scene, and use your student number as the file name. Put the file into the folder J:\SM3122\submit\Week07\ at the end of this class.

** Week 07 End **