Bournemouth University

Media School (NCCA)

MSc Project undertaken in partial fulfilment of the
requirements for the MSc in Computer Animation and Visual Effect

Fire Whirl Simulation
by
Magdalene Chan
i9551473/4053092

Project Unit: MSc Project
August 2009

Abstract

This project introduces and demonstrates the production, algorithm and technique of creating fire whirl. The final product will be some simulation clips about fire whirl, including the application to a scene, and the built Fire Whirl User Interface.

The primary aim of this project is to experiment the effective way of creating fire whirl, which had introduced in Personal Inquiry Unit. The result of this project will focus more on the animation and the shader used for the fire whirl. The success of this project depends on how close this fire whirl to the reality, as well as the implementation of this fire whirl. The secondary aim is to demonstrate the programming and animation techniques learnt throughout the whole course, how the learnt knowledge apply into this project, by building the User Interface for the fire whirl creation. This UI is made for animators, so that they can model a fire whirl easily and apply to the relevant scene.

The main research topics for this project are particles animation along the curve, the fire, fluid and vortex/whirl properties, shader and MEL.

Content

Chapter 1 Introduction

Fire whirl, or fire tornado is a rare phenomenon, which is not easily seen. Its size can be small to fit in a room, or very huge that can be observed in the serious forest fire, depending on the amount of air temperature and current in the formation of the tornado. It is not considered as a kind of tornado, but it can cause significant damages towards the nature. Scientists had done some research on fire whirl, in order to find a solution for extinguish it when it occurs in the forest fire. A example of the scientist research of fire whirl is Experimental Study on the Model Fire Whirlwind (Qin Jun, Liao Guang-xuan,Wan yu-tian, Fan Zhi-hang, 2002).

Fire whirl does not last long, usually appears for a few minutes, unless the wind is strong enough. According to its appearance, it has the image of fiery, strong, aggressive and rapid. As it is rare to happen, it seldom appears in the realistic live action movies. Nevertheless, it sometimes involves in animations (especially in Japanese animation), films or/and games with magical or sci-fi genre. In magical animation or game, fire whirl is either an attack of a sorcerer/wizard/robot, or using for impressive character appearances (provided that those characters' property is fire). These are the example of the fire whirl used in the character scenes:

Fig 1.1 *The firey whirl effect used in the robot attack*

Picture source:

http://www.youtube.com/watch?v=pCxO3Nczoys

In 5-6 minutes

Fig 1.2 *The firey whirl effect used in a online game as the final attack*

Picture source:

http://tjgame.big5.enorth.com.cn/system/2008/11/23/003796232.shtml

The idea of this project is inspired from these pictures. The effect of fire whirl strengthens the characters' images.

It is difficult to create a fire whirl in real life, as it requires a large area according to the video research in YouTube, other conditions had been mentioned above. In addition, it is not a common phenomenon use in the animation. Fire whirl tutorials, plug-in or other scripts/software regarding to the fire whirl creation are not easily found. Hence, this project focuses on the creation of fire whirl and the development of the user controls for its creation.

1.1 Project Objective

This Project aims:

- To understand the methodology and algorithm of creating fire whirl
- To experiment the suggested method in the PI research
- To demonstrate fire whirl simulations
- To study more on creating particles animation and user control using Maya and MEL
- To practice what had been learned in this course, such as the use of MEL and programming algorithm

1.2 Project Specification

Project Title: Fire Whirl Simulation

Project Output: Short Clips which demonstrate fire whirl

Project Format: Experimental Simulation Clips (avi/mov)

Key Research Topic: The methodology and technique of creating fire whirl in Maya

Research Methods: Documentary and Video Research

Key Technologies/ software to be used:

- Autodesk Maya 8.5/2008
- Mel script

1.3 Target User

The product of this Project is targeted to effect and game animators, who need to make fire whirl for their animation, game or other multimedia project. The user requires knowing basic Maya interface, and some knowledge regarding to fire and whirl structure. Additionally, there is only numeric adjustment for colour, so the user needs to know how to calculate colour as well.

1.4 Schedule

The bigger image of the project schedule can be seen in the Appendices section.

More time had been spent on research in the early stage for determining the fire whirl information-- the vortex structure, its animation, appearance, algorithm and the most suitable software and programming language used. Fire whirl skeleton (model) and the user interface are produced simultaneously in the production stage. After experimenting a stage, for example building the curls and the vortex circle, MEL is used not only for creating the user interface, but also to record the production stage by recreating the curls and the vortex circle in mathematical way. This can speed up the production, as well as easily arrange the MEL structure. The project production is processed according to the planned schedule so far. In August, it is the post-production stage for amending errors, further improvement of the interface and rendering the final simulations.

Chapter 2 Previous Work

There are not many experimental CG fire whirl seen in the Internet research, but some real fire whirl experiment and fire whirl disasters can be found in You Tube.

Fig 2.1 *the fire whirl sequence in the laboratory*

Fig 2.2 the fire whirl occurs in the forest fire

Picture Source: http://img1.qq.com/news/pics/17354/17354879.jpg

Fig 2.3 *the CG fire whirl test made by 3Ds Max plug-in—Fume FX*

Picture source: http://www.youtube.com/watch?v=ce8FJkG-sSs

In the website—Instructables.com, it introduces the method to create a fire whirl, with the help of some tools such as tornado chamber and gasoline. The idea of fire whirl is to make the fire motion under control of an air tornado vortex. The air tornado vortex is formed by rapid rotation of air (wind). The gradual increase speed of vortex rotation creates the kinetic energy that is enough to cause the fire follow the vortex motion. In order to make a convincing fire whirl, the understanding of fire and vortex properties are required.

2.1 Fire

Many tutorials for fire creation are found in the book and internet research. Basically, fire can be formed by either in particles approach and fluid approach. In Maya and 3D Studio Max, the fire effect is produced using particles approach. The particles clouds are presented when playing the model in the scene. On the other hand, in Houdini, the fire effect is created using fluid approach. The fluid container appears once the user presses the fire effect (a.k.a. Burst into flame). Fire is produced by the combustion of material, which result in fluid motion. Using fluid approach will give a more realistic fire, but this approach has limited control of fire flow, and also it requires greater memory RAM and graphic card for rendering, due to the complex calculation for the fluid motion. The well-known equation for computing fluid motion is Navier- Stokes equation. It presents the fluid motion change in time under Newton's second law—F=ma. This equation also estimates that the fluid pressure has the relationship to viscosity amount. According to the fire creation tutorial (such as creating fire ring, fire along a wire), particle approach is introduced for controlling the fire motion along an object or an animation path. Unlike fluid approach, it does not have force control while creating. The force(s) have to be inserted and linked to particles manually. The viscosity of the fire is influenced by the particles' amount and the distance between particles. The transparency and the size of the particles can be adjusted by changing the ramp values of radiusPP and opacityPP, or applying linstep/smoothstep algorithm to them. The fire is following the vortex path when the fire whirl is generating, so it would be better to use particle approach. However, fluid approach cannot be omitted as it provides better viscosity and fluid flow control than particles approach. With the reference case (CG talk, 2009) of creating fire using particle clouds with fluid shader, applying fluid shader with particle sampler info into particle cloud would be the best solution in this case. There are few clips about the fire creation using fluid shader in particle clouds found in YouTube. The following screenshot is one of the examples:

Fig 2.4 *firey effect using fluid shape in particle clouds*

Picture source: http://www.youtube.com/watch?v=FM9ag6w4QY0

2.2 Vortex

In Maya, 3D Studio Max and Houdini, they provide vortex field/force control for users to adjust the spinning/rotating motion. It benefits to random particles scattering on the scene. When the vortex field is applied, the random particles will move in the vortex path. A tutorial about vortex field (Maya) is written in Autodesk Maya : The special effect book.(Autodesk, 2009) The particles (placed in the cross pattern) move in a spiral way to make the galaxy scene. Vortex field in Maya can use for both spiral and volumetric vortex movements for the random particles. 3D studio Max also has vortex field to do the same thing. The following picture is a picture of the fire whirl experiment (according to PVadmin's (2009) tutorial) with 3D studio Max vortex field and wind field:

Fig 2.5 *Fire Whirl created in 3D Studio Max*

In Houdini, vortex force control is provided in this way:

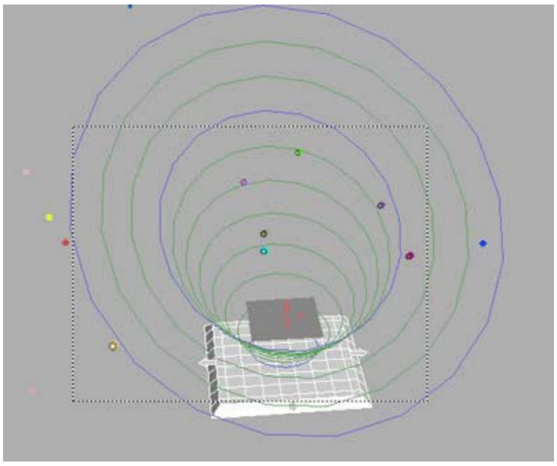


Fig 2.6 *Vortex Example from Houdini official website*

Unlike Maya and 3D Studio Max, it represents the force with the vortex path, Nurbs circles with increasing radii. However, these vortex fields/forces has the same limitation—fixed vortex path and limited controls. Houdini is user-friendly that it allows the user to adjust the radius, velocity, direction, distance, drag, density and other relative forces. 3D Studio Max let the user change the length, the vortex shape (Taper change), speed, direction (clockwise/anti-clockwise choice) and multiple damping forces. In Maya, the user only can change its Transform attribute, magnitude and attenuation. The numbers of vortex curl lines, the vortex scale and the vortex height cannot be altered. The user is unable to modify a desired vortex shape by these existed controls. These controls, together with the missing controls, inspire the development of the Fire Whirl User Interface.

Another way to create vortex (tornado) is to create a helical model and connect the particles to that object using particle goals in Maya, according to a tornado tutorial by Michael McKinley (2009), but this way also has the same weak point—unchangeable vortex path and height. However, this tutorial provides a method for tornado spinning movement. To make the tornado model spin in snake motion, it can be done either attaching a lattice or bone to the tornado model.

2.3 Previous research in Personal Inquiry

Fire whirl research had been done in Personal Inquiry unit. This research contains both fire and vortex properties introduction, which are useful in this Project. These properties information helps to understand the controls, especially shader controls. Those experiments recorded in the Personal Inquiry website: http://nccasymposium.bmth.ac.uk/Chan_Magdalene/index.html

The methodology for making fire whirl is determined from these experiments, including the software and the programming language mainly uses. On the other hand, these experiments arise the problems in making the fire whirl. These problems are necessary to look into and solve in this Project.

These are the problems found during the research in Personal Inquiry:

- Fire shader; from 3D studio Max experiment on making a fire whirl, billboard shading is not advisable to use for shading fire.
- Particle travelling along the curve; Curve emitter cannot get the effect that the particles gradually increased while travelling along the provided path.
- Rendering time Vs shader; using Maya Fluid and Houdini Fluid for fire shader will take a long time to render.

Chapter 3 Technical Background

Fire whirl is the combination of fire and whirl/vortex. It obtains the properties of these elements. Hence, equations such as Vorticity and fluid algorithm have to be used in order to achieve the realistic animation. The colour ramp and the vortex motion are based on the real fire whirl research shown in the last chapter (fig 2.1).

3.1 Algorithm

A. Vortex Algorithm

Complex calculation of vortex is found in the article, which is linked from Wikipedia, under the topic Vortex (Anonymous, n.d.). According to the article, vortex is related to angular velocity:

$$\omega = d\theta/dt$$

Fire whirl animation depends on how fast the particles (fire particles for this project) spin in the vortex circular path, gaining enough energy to whirl up and form the fire tornado/whirl. Applying this equation to this animation, **dt** equals to the total seconds the animation last. **dθ** is set to 90 degrees, that is the vortex model will rotate 90 degrees in more than half seconds:

ω= dθ/dt → 90 degrees/ 6 seconds= 15

The keyframes will be placed with the difference of 15.

In 3.2 Cylindrical Flow of that article, it mentions that "the magnitude of the velocity decrease with the increasing r". This theory can be obviously seen in this Project. The following diagram shows the comparison between the fire whirl with the radius 3 and radius 1:

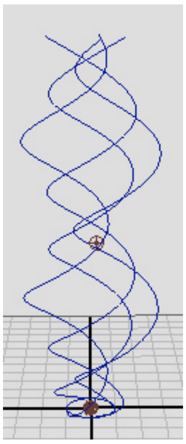


Fig 3.1 *Fire whirl Model with radius=1*

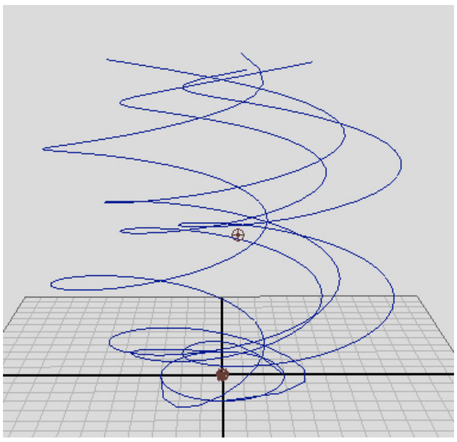


Fig 3.2 *Fire whirl Model with radius=3*

In Fig 3.2, the formation of the vortex paths explains the reason why greater radius has lower velocity. When the vortex circle radius increases, the width (or the radius inside the curl) of the vortex paths become bigger as well, but the height is decreasing even the value of these two model are the same (height is 4). It is because the particles need to take more time and energy to travel the whole circle with greater radius. In this example, the particles may take only half second to travel the whole circle in Fig 3.1 before whirl it up to fire whirl, whereas in Fig 3.2 the particles may need 1 to 2 seconds before whirling up to fire whirl. In addition, with the increasing circle radius, the lengths of the paths are growing longer as well. This is another reason why the particles velocity will decrease due to the size of the circle radius.

Bernoulli's Principle and Helmholtz's three theorems had been visited during Personal Inquiry Research. These theories explain the behaviour of the vortex in details. By reviewing these theories, this proves the accuracy of the project's model.

Helmholtz presented three theorems regarding to three-dimensional motion of fluid in the vortex paths (Kuethe and Schetzer, n.d.):

1. **The strength of a vortex filament is constant along its length.**

   In this project, the velocity of the fire paths are defined with this formula in Maya (adapted from MEL Scripting for Maya animators, Mark R. Wilkins & Chris Kazmier, 2003):

   Goal U=age*Fire Flow Speed – Fire Flow Speed Offset (this will be explained in later section)

   The particles keep moving with this formula from the starting point of the path towards the end of the path. Goal V is set as the same velocity to Goal U. It does not vary with time, or frames

2. **A vortex filament cannot end in a fluid; it must extend to the boundaries of the fluid or form a closed path.**

   This theory indicates that the paths cannot end before the fluid. The problem encounters in this fire whirl is opposite—the fluid (particles) stops on the half way of the vortex path due to the insufficient speed. Inputting greater amount of Fire Flow Speed helps to generate more particles, making it to travel the full path. (it is regarding to Particles-based fluid algorithm, which will be explained later)

3. **In the absence of rotational external forces, a fluid that is initially irrotational remains irrotational**

   When there is no animation applies (keyframe set) to the vortex circle, the fluid will only flow along the fire path without any rotation in this model.

Relevant theorem to Helmholtz's theorems is Kelvin's circulation theorem. It is introduced in the vortex calculation article (Anonymous, n.d.). Kelvin's theorem proves the fluid behaviour in the vortex mathematically based on Helmholtz's theorems.

Bernoulli's Principle is known as—"an increase in the speed of the fluid occurs simultaneously with a decrease in pressure or fluid's potential energy" (Wikipedia 2009 & Daniel Bernoulli, 1738) The equation shows in this principle is incompressible flow equation.

$$\frac{V^2}{2} + G + \frac{\text{pressure}}{\text{density}}$$

where G is gravity, V is fluid flow speed

The result is constant fluid flow, the fluid shape is incompressible regardless of the change on pressure. The density is constant. The density setting remains constant in the Fire Shader for this model. In the Fire Fluid Texture, the colour is interpolated with the change of pressure. There is no influence on colour when editing the output colour value.

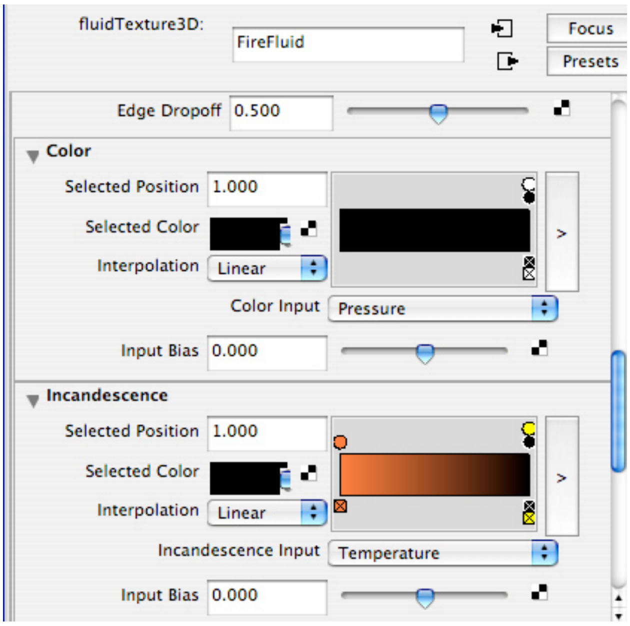


Fig 3.3 *Colour input under the influence of pressure in Fire Fluid (Fire Shader)*

The formula of calculating the vorticity is included in Personal Inquiry research.

Vorticity=Curl*Fluid velocity

In fire whirl, cylindrical tornado shape in the whirl is formed when the fluid flow go faster in the vortex path. The fluid become denser, and 'stick' together easily to produce cylindrical tornado.

Trigonometry is used to calculate the vortex paths. The values in x and z coordinates are defined as follow to plot the helix path:

$z=($Circlerad*2)*cos($i);

$x=($Circlerad*2)*sin($i);

$y=$i;

The formula ($Circlerad*2)*sin($i) is the variation of Rsin (θ) (the z coordinate will be Rcos (θ)).

X=Rsin(θ)

Y=Rcos(θ)

where R is the radius in the circle, θ is the angle

This formula is found in this book: Computer Graphics, Mathematical first steps (P.A, Egerton & W.S. Hall, 1999)

It is a common formula for computing a circle. This formula also uses to allocate the curve in a circle, like this in CreateVortex():

$cx=$Circlerad*cos($angle);

$cz=$Circlerad*sin($angle);

….

move $cx 0 $cz

and bone animation (AnimateBone() in the script):

$Spinx=$SpinRadius1*cos($Spinangle);

$Spinz=$SpinRadius1*sin($Spinangle);

setAttr $JnameMoveX[$i] $Spinx;

setAttr $JnameMoveZ[$i] $Spinz;

The bone will move under these formulae, which are inserted in the translate X and Z of the bone.

From the formula ($Circlerad*2)*sin($i)/ ($Circlerad*2)*cos($i), it is known that:

- The radius of the helix is twice of the vortex circle, $Circlerad is the variable of vortex circle radius.
- the helix move upwards, in y axis.

In Mathematics first steps book (P.A, Egerton & W.S. Hall, 1999), the equation of the circular helix is introduced as :

r (θ) =*R* cos (θ) **i** + *R* sin (θ) **j** + Cθ **k**

Normally, this formula is expressed as r (θ)=x + y + z, but in this project, this formula is defined in this way: r (θ)=z + x + y

Cθ represents increasing angles. $y is declared as parameter $i, which angle will incrementally increased in y axis while plotting the helix.

B. Fire Algorithm

Fire is gaseous phenomena, as introduced in MSc Fire and Smoke Simulation Project in 2007 (David Minor, 2007). It is computed using Navier-Stoke equation. It is important to define if fire whirl is compressible before computing fluid. In the last section, it mentions that fluid is incompressible in the vortex, however, fire is stated (Rick Parent, 2008) as compressible. Since fire is flowing along the vortex paths in the fire whirl process, it obtains the properties of vortex, which considers as incompressible fluid. Nevertheless, by observing the motion of fire whirl, it is not totally incompressible:

Fire compression

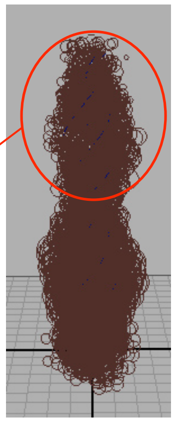


Fig 3.4 *The compression during the fire whirl process in real life.*

Fig 3.5 *The simulation of the fire whirl compression from the real life picture (shown in particles distribution mode in Maya)*

Although it obtains vortex properties, the fire properties is still presented, especially the compression part shown in Fig 3.4 and 3.5. Therefore, fire whirl is considered as partially compressible gas phenomena.

In Computer Animation Algorithm and Technique (Rick Parent, 2008, short form as CAAT in the following paragraphs), under the topic 8.2.1 –General Approaches to Modelling Fluids, there are three approaches had been demonstrated for modelling Fluid:

1. **Grid–based methods (Eulerian formulations)**

Two diagrams are illustrated in CAAT (Rick Parent, 2008) for explaining the fluid (gas in the diagram) flow in and out in a cell and a grid.

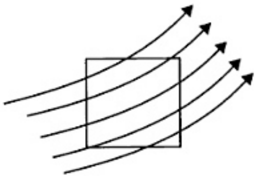


Fig 3.6 *(diagram from CAAT) Gas flow through a cell*

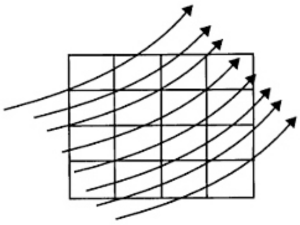


Fig 3.7 *(diagram from CAAT) Gas flow in a grid of cell*

The fluid will be updated in the cells by time steps. When the gas passes by the cells, the cells determine the illumination and visibility of the gas, storing these information and generate the effect through rendering. The velocity, acceleration and other gas properties are tracked while the gas travels from cell to cell. The example of this is the 3D container (without emitter) in Maya:

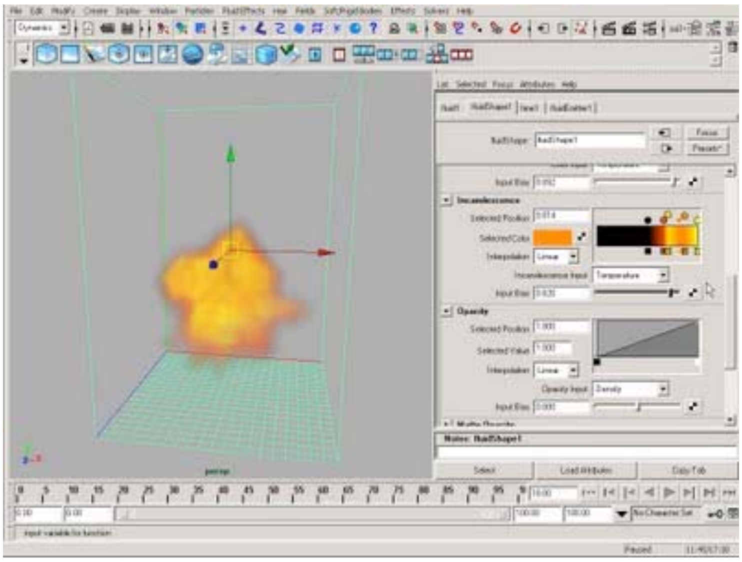


Fig 3.8 *Explosion created using Fluid 3D container in Maya*

Picture source:

http://bp3.blogger.com/_1NTS9B7oDek/SE6SGVQqrFI/AAAAAAAAB2Q/yU6dm0xYPZ0/s400/dsc03a.jpg

It is considered as Non-dynamic fluid effect, according to The Special Effect Book (Autodesk Maya, 2009), because this method do not use fluid dynamics equations to generate fluid effect. In this case, rendering fluid using this method is the quickest among the three methods. The disadvantage of this method had been explained in an articles called Smoothed Particles Hydrodynamics (Anonymous, n.d.). It point out the weakness of grid method:

a. The movement of the material cannot be tracked in the grid, which cause the difficulty to control the time step of the fluid animation.

b. The deformation of the material is limited. To generate irregular geometries of material, expensive numerical mapping may be required.

c. The detection of the fluid material deformation and motion may not be accurate.

d. More calculation on the grid may be required to simulate more accurate render data.

2. Particles-based methods (Lagrangian formulations)

With the reference of the method name, the fluid effect is created by the update motion of the particles. The particles can be rendered in spheres, blobby surface, cloud etc. This project generates the fire whirl mostly based on this method. The computation process of this method is simpler as its calculation is similar to rigid body dynamics and Newton's 2^{nd} law.

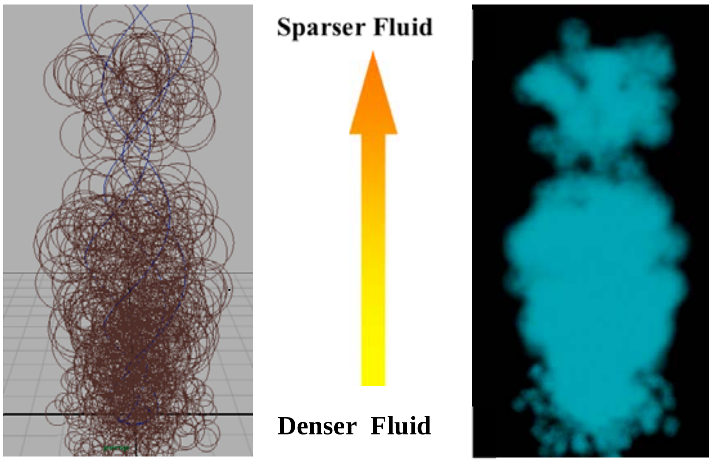


Fig 3.9 *Wireframe generation of fire whirl* ***Fig 3.10*** *Particle Cloud shade of fire whirl*

Its disadvantage is that more particles are needed to produce denser Fluid. Fig 3.9 and Fig 3.10 demonstrate the distribution on the fire particles in the fire whirl (the beginning of the simulation) in wireframe and particle cloud shade respectively.

Other examples of fluid simulation using this method are Maya Fire Effect and Smoke Effect.

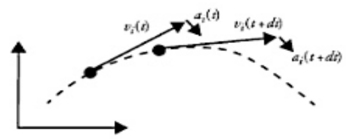


Fig 3.11 *(diagram from CAAT) the animation update of particles-based method fluid*

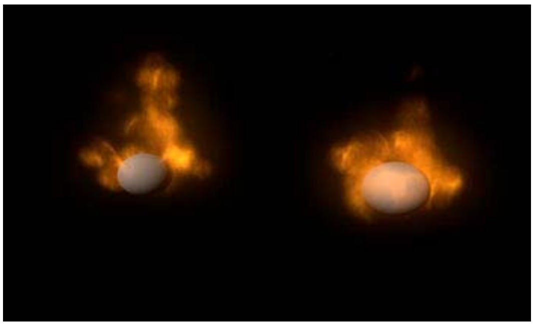


Fig 3.12 *Fire simulation using particles generation method (left) and Fire Effect (right)*

3. Hybrid methods

This method is the combination of Particles-based and Grid-based methods. It includes the algorithm of both methods, making use of the advantages of both methods to simulate fluid. The particles deal with the deformation and the motion of the material and the grid takes care of the visual and illumination of the fluid. Maya Fluid Effect with emitter is the example of simulating fluid using this method. The rendering is the most time consuming, but the resultant effect is the best among three methods.

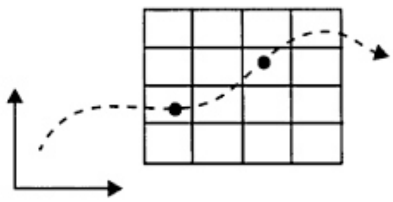


Fig 3.13 *(diagram from CAAT) the animation update of hybrid method fluid*

It had been mentioned that this project is using mostly particles-based approach to achieve the fire whirl motion. On the other hand, Hybrid method is partially used as well, in order to obtain better visual fluid effect. The grid-based method is used in Fluid Texture in the fire Shader.

The incompressible fluid dynamics equation (Navier-Stoke equation) had been briefly presented in Personal Inquiry Research (the formula image is taken from Wikipedia):

$$\overbrace{\rho\Big(\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Unsteady acceleration}} + \underbrace{\mathbf{v}\cdot\nabla\mathbf{v}}_{\text{Convective acceleration}}\Big)}^{\text{Inertia (per volume)}} = \overbrace{\underbrace{-\nabla p}_{\text{Pressure gradient}} + \underbrace{\mu\nabla^2\mathbf{v}}_{\text{Viscosity}}}^{\text{Divergence of stress}} + \underbrace{\mathbf{f}}_{\text{Other body forces}}.$$

In the reference on the vortex theory and principle, this equation should be used for this Project. Pressure gradient and viscosity are solved in the fluid 3D texture. The detail of the Fluid Shader will be explained in the shader section. For viscosity, it can also be achieved by adding the attributes—radiusPP and OpacityPP and adjusting them using ramps.

The acceleration of the particles can be modified by inserting this formula to lifespanPP:

lifespanPP=(dnoise(position x 0.5 x scale+(-3) x time)+0.9) x lifespan;

This formula is adapted from Maya Fire effect. With the function of “dnoise”, the random movement between particles can be achieved. The compressible effect of the fire can be adjusted by the added number “0.9” in the formula. This number adjusts the height of the compression, the value 0.9 means the compression takes place near to the top of the fire whirl. The lower the value, the compression will take place in the lower part of the fire whirl.

With the combination of this dnoise function with the fluid dynamics equation

Lifespan is defined as the same value as vortex circle radius. For the velocity and the acceleration of the fire, it can be obtained by this equation, which had been early mentioned:

Goal U=age*Fire Flow Speed – Fire Flow Speed Offset

Particles goal is used in order to achieve the similar effect as Maya Curve flow effect.

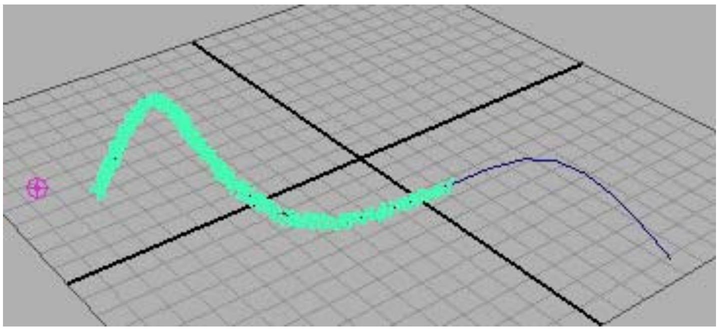


Fig 3.14 *Particle goal animation*

The fire whirl animation seen in the reference clips (YouTube, 2009) show that it is generated by the gradual increase of the particles quantity and the updated velocity and position of particles. Particle goal is the only method to get this result. The amount of the particles can be adjusted by adding this formula to goal Offset:

goalOffset=sphrand(VR*2)

VR is Vortex Radius, which is user-defined.

Sphrand is the sphere random; it produces particles randomly in the spherical volume.

Goal U and V has the same formula to get the constant particles speed in the whirling process. Fire Flow Speed and Fire Flow Speed Offset is user-defined. There is a better explanation of Fire Flow Speed and its Offset in User Interface Guide, please refer to the Appendix A1.

This if statement needs to be added in order to prevent the particles accumulate at the end of the curve:

if (goalV>FS && goalU>FS)

{lifespanPP=0;}

FS is Fire Flow Speed, which is user-defined.

If the particles reach Fire Flow Speed point, the particles will be deleted.

Vortex circle particles have slightly different algorithm from fire path particles:

lifespanPP=5;

goalU=age*VR*20-2;

goalV=age*VR*20-2;

goalOffset=sphrand(2)

The particles velocity depends on Vortex circle radius, unlike fire path particles animation, which velocity is user-defined.

External force such as Drag, Gravity and Turbulence are added to adjust the particles animation and colour. In Turbulence Phase Y, a formula is added for the improvement of the turbulence in Y axis. This is adapted from Mastering Maya 8.5 (Autodesk, 2008).

phase Y=noise(time);

3.2 Code Structure

This simulation is written in MEL, which has the similar coding structure as Maxscript. According to BSc Computer Animation Year 3 Tools for Animation Project, under Essay B, 3Ds Max has similar structure to C++, such as function, loops, objects and classes. Maya is originally a C++ software product, so those components mentions in 3Ds Max can be found in Maya as well. This can be proved by listing the attributes, function and nodes (excluding shaders) used in this Project.

Table 1 Node/object and function use in this MEL script:

Node/Object	Functions Uses
NurbsCircle (Vortex Circle)	vortexCircle (), CreateVortex(), AnimateVortex(), DeleteVortex(), SelectVortex(), DeleteKey(),SetFireFreeKey(), fireOtherAction(int $type1)
Curve (fire path)	Helix(), CreateVortex(), FireAnimate(), DeleteFireAnime(), DeleteVortex(),
Emitter and particles (fire)	CreateVortex(), FireAnimate(), SetFireFreeKey()
Emitter and particles (smoke)	SmokeEmit(), DeleteSmoke(), SmokeAnime(), DeleteSmokeAnime(), DeleteVortex(),
Forces (fire): Drag, Gravity and Turbulence	addForce(), DeleteVortex(),
Forces (smoke): Uniform and Vortex	SmokeEmit(), DeleteSmoke(), DeleteVortex()
Joint (bone)	Addbone(), Deletebone(), Animatebone(), DeleteAllBoneAnime(), SelectBone(), KeyCurrTime(int $type0), DeleteVortex(),

Those functions are not stated in this table are either shaders related (such as fireshader(), Smokeshader(), DeleteShader(), DeleteSmokeShader()) or functions related to the availability of the user controls.

All the components are parented to vortex circle in the script, for better selection and deletion in the scene. Hence, the node VortexCircle appears frequently in certain functions.

The loops which are mostly use are for loops, switch loops and if statement. If statement uses for asking the program to execute certain function if a variable reach this value.

Example on if statetment:

```
In DeleteVortex():
int $Boolean=`checkBox -q -v EnableSmoke`;
if ($Boolean==1)
 {
   select -r smokeramp;
   checkBox -e -v 0 EnableSmoke;
   floatFieldGrp -e -en 0 SmokeUni;
```

```
  floatFieldGrp -e -en 0 SmokeVortex;
  doDelete;
 }
```

The smoke related control will be disable if the smoke check box is on while DeleteVortex() executes. The smoke ramp will be deleted as well.

For loops are mostly used with editing attributes and selecting for fire paths, emitters and particles.

For loops example on curve selection:

```
for ($i=0; $i<$CurveNum; $i++)
 {
   $curvename[$i]="curve"+($i+1);
    select -add $curvename[$i];
 }
```

String array needs to be made for the names (usually for multiple selection and edition) before making selection in for loops. It is very commonly use in MEL script for object names and attributes' names (Mark R. Wilkins & Chris Kazmier, 2003).

Switch loops are usually used for check box enable/disable function, and buttons for changing values. The cases in the switch loops are 2, except the edition of transform attribute:

```
global proc fireOtherAction(int $type1)
{
 //*****switch function for setting Fire Whirl Transformation*****
 switch($type1)
 {
   //*****type1=1, Fire Whirl Translation can be controlled  by user. *****
   case 1:
   float $firepos[];
   $firepos[0]=`floatFieldGrp -q -v1 FirePos`;
   $firepos[1]=`floatFieldGrp -q -v2 FirePos`;
   $firepos[2]=`floatFieldGrp -q -v3 FirePos`;
   setAttr "VortexCircle.tx" $firepos[0];
   setAttr "VortexCircle.ty" $firepos[1];
   setAttr "VortexCircle.tz" $firepos[2];
   break;

    //*****type1=2, Fire Whirl Rotation can be controlled  by user. *****
    case 2:
   float $firerot[];
```

```
  $firerot[0]=`floatFieldGrp -q -v1 FireRot`;
  $firerot[1]=`floatFieldGrp -q -v2 FireRot`;
  $firerot[2]=`floatFieldGrp -q -v3 FireRot`;
  setAttr "VortexCircle.rx" $firerot[0];
  setAttr "VortexCircle.ry" $firerot[1];
  setAttr "VortexCircle.rz" $firerot[2];
  break;

  //*****type1=2, Fire Whirl Scale can be controlled  by user. *****
  case 3:
  float $firescale[];
  $firescale[0]=`floatFieldGrp -q -v1 FireScale`;
  $firescale[1]=`floatFieldGrp -q -v2 FireScale`;
  $firescale[2]=`floatFieldGrp -q -v3 FireScale`;
  setAttr "VortexCircle.sx" $firescale[0];
  setAttr "VortexCircle.sy" $firescale[1];
  setAttr "VortexCircle.sz" $firescale[2];
  break;
  default: break;
}
}
```

Global and local procedures are used for declaring functions. In this Project, global procedure are considered as the functions which are supposed to execute in the user interface, whereas local procedures are the sub-functions which will multiply use in global procedure functions.

Owing to the great amount of for loops, the parameters (I,j,k,l,m,n,p,q,s) are declared as local variables in the beginning of the script.

The history (process) of creating this script had been written in the MEL script itself. 10 version (including the final one) had been created in the process. The previous versions can be seen in the CD. A diagram, which is related to code structure, can be seen in the CD as well.

The following table are the Node and attributes list used in this program:

Table 2 Node/object and attribute list in this MEL script:

Node/Object	**Attributes Uses/Exist**	**New Attributes added**
NurbsCircle (Vortex Circle)	Radius, name , Translate XYZ, Rotation XYZ, Scale XYZ	None
Curve (fire path)	Position, knot, degree , name, Translate XYZ, Rotation XYZ	None
Emitter and particles (fire)	Emitter rate, lifespanPP, lifespan, Render type, lifespanMode, name, Speed, Speed Random	Particle cloud radius, goalU, goalV, goalOffset, radiusPP , Vortex Radius, Flow Speed, Vortex height, Flow Speed Offset
Emitter and particles (smoke)	Emitter rate, volume Shape, away from Axis Along Axis, Scale Rate by Object size, lifespanPP, lifespan, lifespanMode, lifespanRandom , Render Type, Translate Y, Scale XYZ, name	radiusPP, Particle cloud radius
Forces (fire): Drag, Gravity and Turbulence	Magnitude, Attenuation, direction Y (Gravity), Phase Y(Turbulence) , name	None
Forces (smoke): Uniform and Vortex	Magnitude, Attenuation	None
Joint (bone)	Position XYZ, zeroScaleOrient, Oriented Joint, secondaryAxisOrient, Translate XYZ	None

3.3 User Interface

The details on the function and the usage of the controls can be seen in Fire Whirl User Interface Guide.

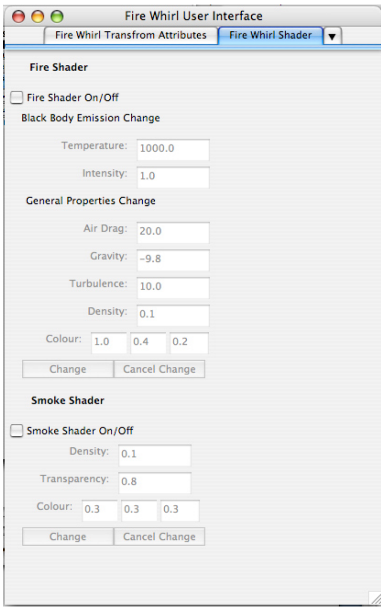


Fig 3.15 *Fire Whirl User Interface*

Fig 3.16 *Fire Whirl User Interface in frame Layout*

The final Fire Whirl User Interface is arranged in tab layout. Originally, its layout is frame layout, but in Fig 3.16, when the User Interface is opened, some of the frames are too long that other frames are unable to see, unless the user resize the window, or collapse all the frames. It is quite troublesome that the user has to collapse all frames before using this UI. It is not very neat to arrange like this. Hence, the interface had been changed to Fig 3.15. The script for frameLayout is easily seen in MEL Fundamentals (Alias, 2004) , MEL Scripting for Maya Animators (Mark.R. Wilkins & Chris Kazmier, 2003) and other MEL tutorials on the internet. However, the tabLayout script can only found in MEL online reference and this website (Tomohiro Abe, 2009):

http://www.not-enough.org/abe/manual/maya/quick-ref.html

This website is in Japanese though, it provides all the UI components with pictures and writing method. The Layout writing for Fig 3.15 is adapted from this website. The making of tab layout is different from frame layout that the commands need to be declared as strings, after that the main command tabLayout calls at the last second line (the line before showWindow), edits and rearranges all the column Layout with tab title. This is the example of tab Layout taken from that website (Tomohiro Abe, 2009):

window;

```
string $tabs = `tabLayout`;
    string $col1 = `columnLayout`;
    button;
setParent ..;
    string $col2 = `columnLayout`;
    button;
    button;
setParent ..;
tabLayout -edit -tabLabel $col1 "tab1"  -tabLabel $col2 "tab2" $tabs;
showWindow;
```

Float/int field group, check box and buttons are mostly used in the UI. Float/int field group are used for making some of the values to be user-defined . The check boxes are used as a switch for some functions, for example the smoke enable in the Fire Animation tab. The buttons are the execution keys for the simulation. The examples of using these UI components can be seen in the above section and the full MEL script.

The UI consists of five frames:

Fire Whirl Creation—to create fire whirl model.

Fire Animation—to animate fire (and smoke)

Whirl Animation—to animate the whirl spinning speed

Fire Whirl Transform Attribute—to move, rotate, scale the fire whirl

Fire Whirl Shader—to add shader(s) to fire and smoke respectively.

3.4 Shading and Rendering

Shader plays a very important part in this Project. The final effect depends heavily on the shader. Besides, the radiusPP and opacityPP ramps affect the particles appearance.

a. Ramps

Ramps are commonly used in adjusting radiusPP, opacityPP and rgbPP. (Autodesk, 2009) rgbPP is used for adjusting individual particles colour. The colour for the fire whirl model is under control by Fire Shader, so the setting of rgbPP can be omitted. radiusPP is the attribute for resizing individual particles; opacityPP is dealing with the transparency of each particles. In MEL Fundamental (Alias, 2004) and the Special Effect Book (Autodesk, 2009), two methods had been introduced for adjusting radiusPP and opacityPP. One of the method is to insert linstep or smoothstep formula , such as smoothstep (0, lifespanPP, age) into the expression. It had been tried, but the result is not effective:

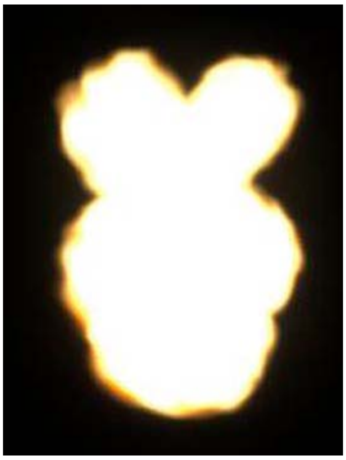


Fig 3.17 *The bad result of using smoothstep and linstep in radiusPP and opacityPP*

Another method, which is used mainly in this Project, is applying ramps (arrayMappers) into the attributes.

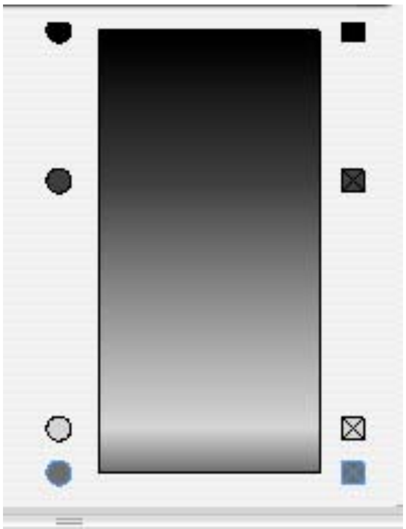


Fig 3.18 *the radiusPP ramp for fire paths' particles*

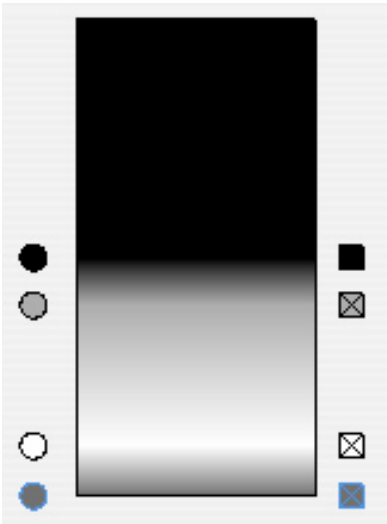


Fig 3.19 *the opacityPP ramp for fire paths' particles*

The ramps in radiusPP and opacityPP are presented in black to white gradient. In radiusPP, the darker the grayscale, the smaller of the particles' size. In opacityPP, black represents transparent, whereas white represents opaque.

The bottom part of the ramps are brighter because there should have more obvious particles in the fire, which is the centre of the heat and light source. After that, the particles are supposed to disappear gradually according to the fire properties.

Using ramps is better than the linstep and smoothstep algorithm as it is easier to adjust the transparency and the size of the particles precisely.

b. Fire Shader

By doing the tutorial on Maya fluid effect taught in the Special Effect Book, it is confirmed that using fluid effect is not possible for this Project owing to the huge time consumption on rendering. However, fluid has to be used to simulate the viscous effect in the fire. Research on the alternative method of creating fluid like shader are found in one of the post (the topic is fluid tornado) and Autodesk online reference, under the topic "Shade the particles with Fluid Shape", the fluid shape is successfully created but the shading data cannot be seen in the render image, unless one of the presets applies to the fluid shape. The algorithm of the presets are wrapped, the methodology of inserting it to the fluid shape cannot be seen. Therefore, another approach is tried based on this fluid shape method. In the Autodesk guide, it demonstrates the creation of fluid shape to the particles in the following steps:

1. Create Cloud particle system, which is setting the particles render type to particles cloud.
2. Assign Fluid Shape to the particles
3. Add contents to the container. In CG Talk, this step is inserting fluid presets to the fluid shape.
4. Edit the fluid shape properties and add particles sampler info to particles attributes.

A tutorial (Alex Alvarez,2009) on creating candles flame had mentioned the details on adding particles sampler info (in that tutorial, the tutor adds sampler info because the candle is a nurbs geometry). The difference of the candle transparency can be seen in this website: http://www.thegnomonworkshop.com/tutorials/candle.html

The shading network for this fire whirl is created using particle cloud as the main volume shader and applies Fluid texture as the colour output. This is the final shading network:

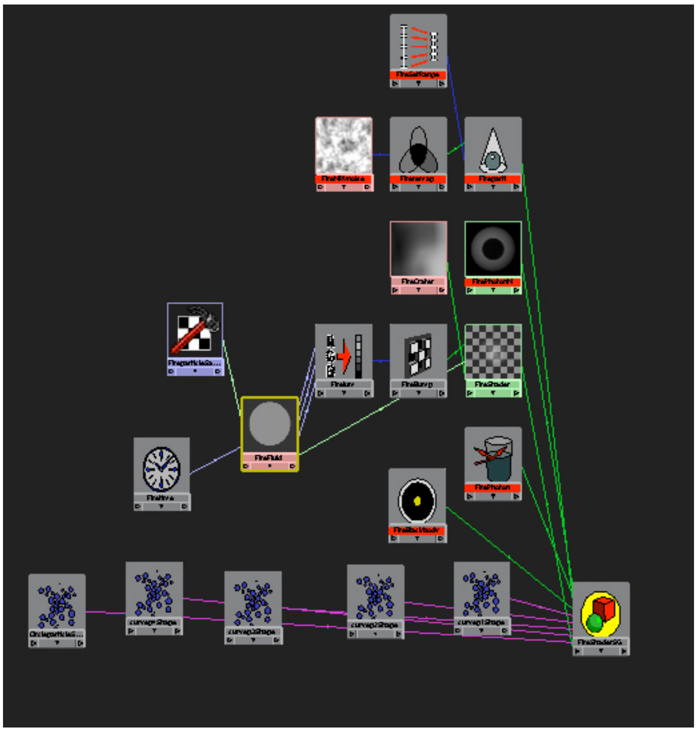


Fig 3.19 *Fire Shader network*

Crater and noise are added into blob Map and noise in the fire shader respectively for adjusting the tone of the fire. The channel colours in Crater texture will affect the output colour of the fire whirl. Adding the particle sampler info to the transparency in both Fire Shader and Fluid Texture enhance the visibility of the Shader. (it appears to be more visible in the shader preview.) The attributes settings are partially adapted from the Gnomon Workshop video on Pyrotechnics and Fluid, and the Special Effect Book (Autodesk, 2009). In Section 3.1, Navier-Stoke equation had been introduced. This equation is applied as a solver to the Fluid texture. The viscosity is allowed to control in Fluid. The pressure is applied as colour input in the shading color for tone down the brightness of the fire, but the fire is still appeared as too bright. With the help of this equation, the colour of the fire varys when it is burning. The Opacity is influenced by Density. Incandescence is affected by temperature. The Fire colour can be changed in the Incandescence ramp. One of the colour Entry can be adjusted by user in Fire whirl UI. Other nodes, which are not mentioned, are the nodes related to Mental Ray Rendering.

c. Mental Ray Rendering

After several adjustments on Fire Shader, the appearance still looks smoky like this:

Fig 3.20 *One of the fire shader test*

According to Personal Inquiry research, Photon mapping and black body are important properties to the fire. In on of the tutorial from CG Arena (Silvia Palara, 2005-2006), it teaches to create firey effect using mental ray shaders. Most of the mental ray nodes set up are adapted from this tutorial. The adjustment of black body is referred from Digitial Tutors video on Mental Rays Nodes (Digital Tutors, 2009). The temperature and the intensity of black body nodes are user-defined. The default temperature is 4600K. To ensure the use of black body, some fire render test with different temperature and intensity had been done.

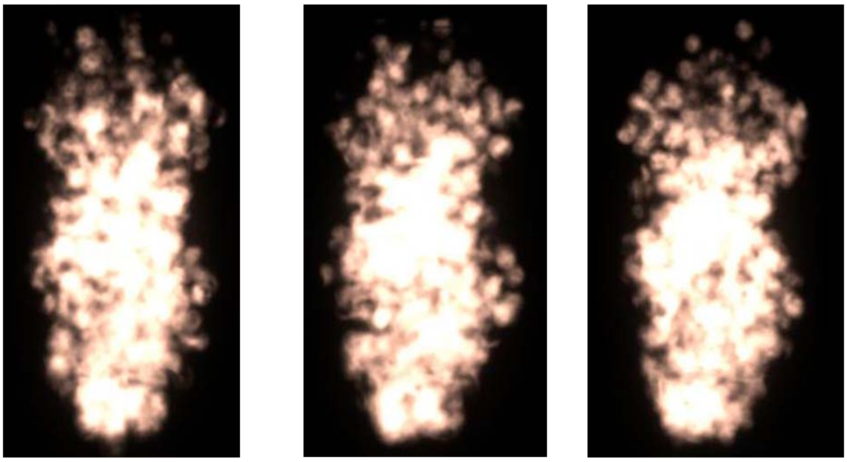


Fig 3.21 *the comparison on different black body temperature and intensity.*
left: 10000K intensity 1 center 1000K intensity 3 right 1000K intensity 1

The divergence and the tone are obviously changed in the left picture. The fire is slightly duller than the other two as 10000K will release blue light. The divergence in the centre picture is changed as well due to the increase of the incandescence, which make the particles stick together more than the left picture (the original image), but for the visual effect, it looks flat among all.

Photon mapping (Parti Volume Photon Shader in the mental ray node) and dielectric material are necessary to add for making the fire looks more like fluid. Dielectric material will give the bolt effect to the fire, that is bright in the centre and gradually change colour from the centre to the edge. The rendering stage will be explained in the Problem Encounter as it is one of the problematic area in this Project.

Chapter 4 Problem Encounter

There are quite a lot of problems encounter in the production process, from the creation of the fire path to the rendering process.

Problem 1 Curve Position

It was found out that the position of the curve is wrong when doing the animation and UI adjustment. In the mid Presentation, the curve paths of the fire whirl is like this:

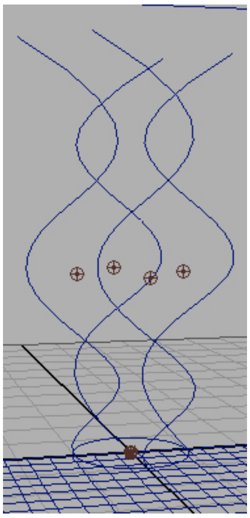


Fig 4.1 *Old Fire whirl curve*

It looks the vortex paths are correct, but when the radius increases, the vortex paths are not intersected with one another.

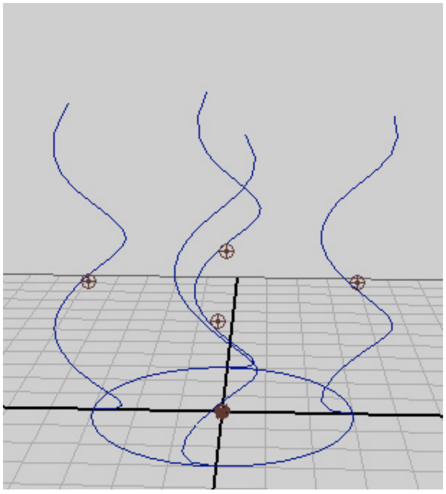


Fig 4.2 *Old Fire whirl curve in circle radius 3*

When the particles generate in that curve, particles will not blend with one another. This means that the model (the curve position) has to be remake. The curves are reallocated with the reference on the vortex images. The fire paths become more difficult to adjust, as the rotation is not constant with different numbers of path. After many tries on the calculation of the path positions, even the positions of coordinates are known that they vary within sin curve, the sin function cannot be inserted in the rotation command. (i.e. rotate 0 sin(time) 0)

The only solution for this problem is to add if statement to every path numbers, and setting the minimum and maximum numbers of the path, so that 4~5 if statements are needed to add to the script. Although it is not the best way to solve the position , the curves' current location looks more logical than before. Fig 3.1 shows the current fire whirl model with the correct paths' position.

Problem 2 Particle Goal and Animation

Goal cannot be directly added to the curve. If so, the animation will look like this:

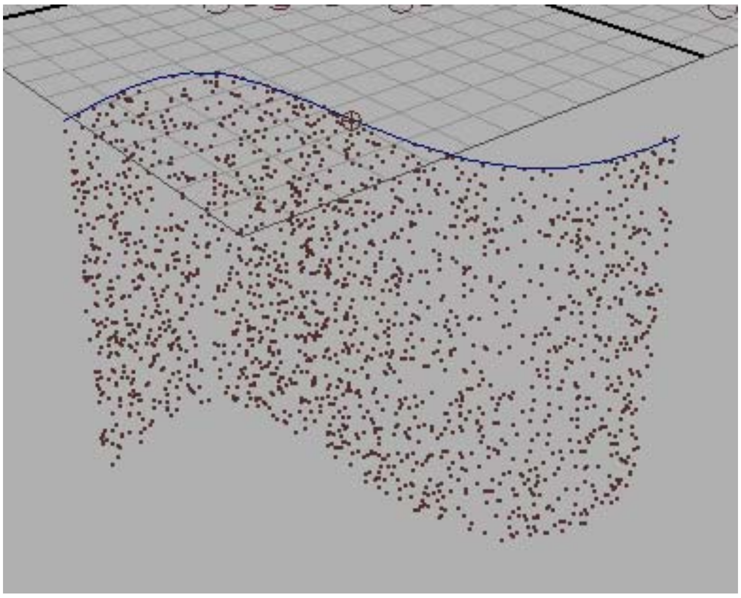


Fig 4.3 *The effect of directly adding goal to the curve*

Fig 4.3 shows the result of this approach. The particles would not move along the curve, they fall from the curve instead. Only adding goal to the curve is effective. Another problem encounter regarding to particles goal is the particles animation along the curve. The speed and the offset value varies to the height and the circle radius. Setting the speed and offset value with the different setting of the fire whirl's height and circle radius had been tried, but the equation cannot perform well in the animation due to the data error. These experiments can be viewed in the CD, under animation_data folder. The final solution to this is to set speed and offset to be user-defined. The user can input whatever value he want to get the desire animation.

Problem 3 Animating Tornado

By reading the tutorial on Michael McKinley Website (Michael McKinley, 2009) , it seems to be a necessary to create the spinning animation for fire whirl segments. The tornado is made by creating nurbs primitive in that tutorial, so the lattice method can be used to obtain the spinning animation. However, this method cannot apply to this model because it is made by curve lines. Another approach for making the spinning animation can be found in Chinese Maya tutorial website (but the reference is lost) . This is binding bone approach. It can be bind skin with the model, therefore it is decided to use bone for the spinning. The bone is the optional function in the user interface.

Problem 4 Fire Shader

The main problem for this Shader is that the fire appears to be too bright. Density, the channel colours in Crater , some Fluid attributes, and even the glow intersity had been modified, the fire still looks very bright. This problem is unsolved upon the completion of the Project.

Problem 5 UI interface

The only problem encounter for User interface is the layout design. The buttons and fields are displaced while opening the interface. These can be overcome by parenting the column layout to the main column/row layout, which is introduced in Morten Moen Website (Morten Moen, n.d.). Other possible solution are adding the width to the button and adjusting the column/rowColumn Layout by adding the flag –columnAttach 1 “left” 15 (example).

Problem 6 Rendering Scene

The first problem meet during rendering is show as follow:

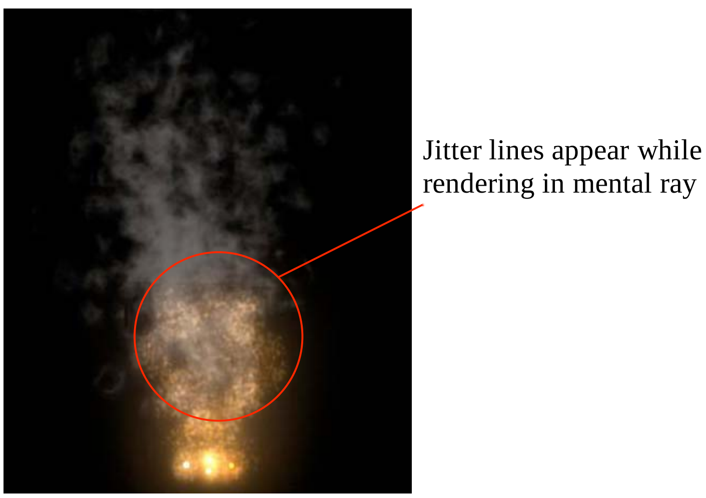


Fig 4.4 *Jitter lines appear in mental rays rendering*

This can be solved by two methods:

1. Adjusting the Max Sampling Level in Raytracing /Scaling
2. Adding Luminance node and link it to the bump map of the shader. This method is written in Autodesk Maya online help (Autodesk, 2009) , under the topic Fluid limitations
3. Avoid using JPG for saving render images.

Another rendering problem is the rendering time. It is unexpectedly longer, though the total rendering time is still faster than using Fluid Effect. The only solution is to separate them in the render layers, render the components individually and composite them.

Chapter 5 Project Outcome

The screenshot of the final simulation are displayed as follow:

Fig 5.1 *the fire whirl with default setting*

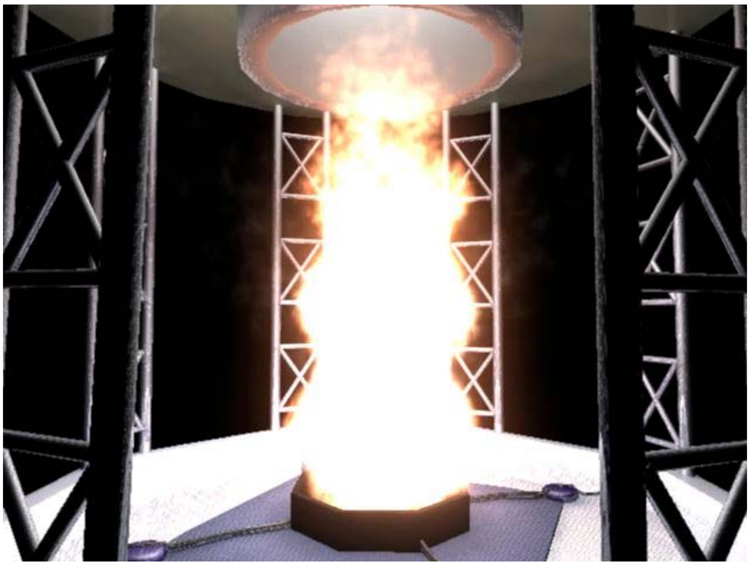


***Fig 5.**2 the fire whirl rendered in a stage using mental ray*

Other rendering approaches for the stage scene:

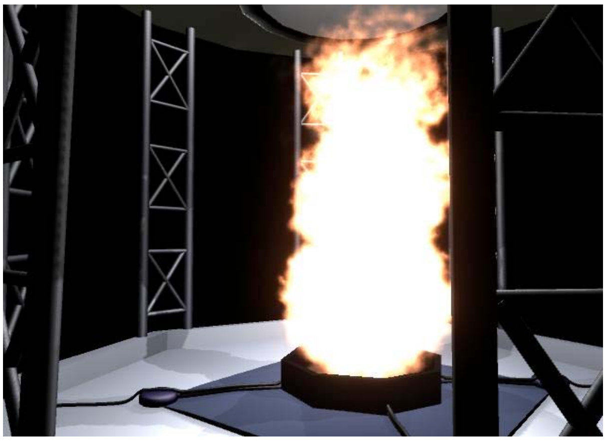


Fig 5.3 *the fire whirl render in Maya software, with 4 spotlight setting in the scene*

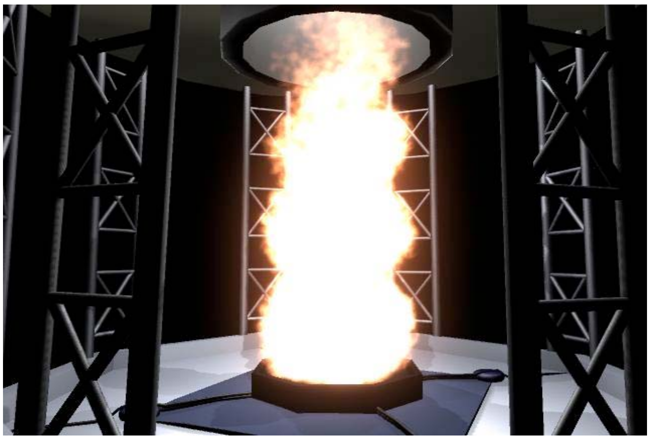


Fig 5.4 *the fire whirl render in Maya software, with 8 spotlight setting in the scene*

From the observation on the rendered screen shot, this shows that Maya software is not suitable for rendering fire whirl with the light setup in the stage scene. In Maya software rendering, the shadows of the particles are being tracked, which make the scene looks awkward. Using mental ray render takes more time, but it provides a more realistic result.

This scene is referenced from David Copperfield Magic show which is shown in Youtube:

Fig 5.5 *The real David Copperfield Magic Show stage*

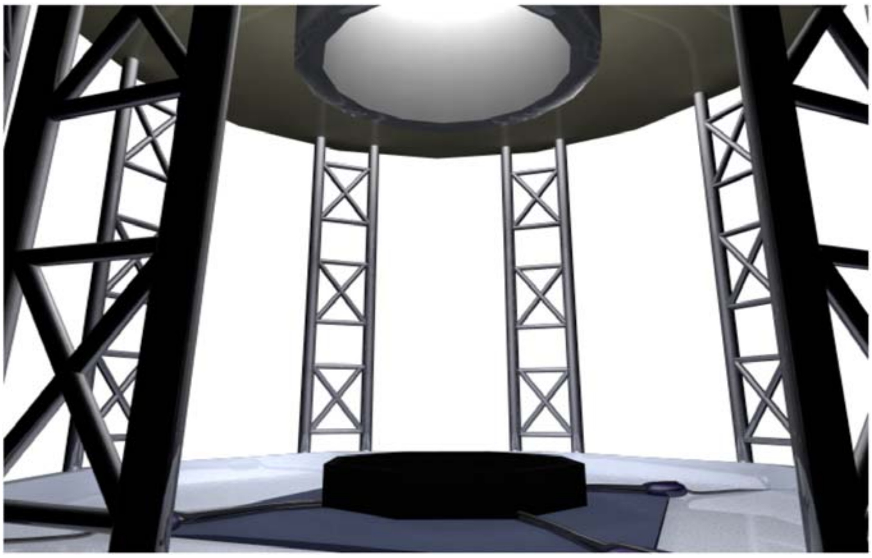


Fig 5.6 *The stage scene with the fire whirl. It is rendered in mental ray.*

The project outcome are the User interface for fire whirl creation and animation and 3 simulations clips—one with default setting, one with colour and position change (which does not shown here) and the simulation with the scene (due to the render error, the animation is rendered up to the highest burning point, the disappearing sequence cannot be shown in the submission, but it may be shown on the website and/or presentation) . Some experimental fire whirl clips are included in the CD as well.

Chapter 6 Conclusion

This is a challenging project overall. The algorithm behind it is quite complicated but most of the project objectives had been achieved. The theory and research stated in the Personal Inquiry unit had been proved through this project. In the process of this Project, the most time consuming parts are the animation and shader adjustments. Three weeks were taken to accomplish them. Writing Maya UI had been learnt more in depth via this Project. Besides, more knowledge on animating and rendering particles in Maya had been obtained. For the final simulation, the original plan is to implement it to a 3D character. Owing to the less time on editing and animating the 3D character, the stage scene is replaced as the final simulation, but the result is quite fine. There should be one more clip for fire colour change, its rendering time cause this clip not able to be one of the submission simulation. This project could be further improved by making the interface be able to create multiple fire whirls in the scene and adjusting the brightness for the fire shader. If possible, making the fire look more volumetric. In the simulation, the volumetric shape of the fire whirl is not very obvious. The appearance and the animation of the vortex circle particles should be improved as well by adjusting its radiusPP and opacityPP.

Reference (Books and Dissertation)

1. Autodesk (2009), Learning Maya 2009: The Special Effect Book, Autodesk Maya Press and Sybex Wiley Publishing Inc.
2. Autodesk (2009), Mastering Maya 8.5, Autodesk Maya Press and Sybex Wiley Publishing Inc.
3. Eric Keller (2009), Maya Visual Effects: The Innovator's Guide, Sybex Wiley Publishing Inc.
4. Rick Parent (2008), Computer Animation Algorithm and Technique Second Edition, Elsevier Inc.
5. Magdalene Chan (2007), Tools for Animation, Artefact Report and Essay Writing
6. David Minor (2007), Physical Simulation of Fire and Smoke Master Thesis, NCCA Bournemouth University
7. Mel Krokos (2007), Texture and Lighting Models Lecture Notes, University of Portsmouth
8. Autodesk (2006), Autodesk 3Ds MAX 9 MAXScript Essentials, Autodesk Inc and Focal Press.
9. Lee Lanier (2006), Maya Professional Tips and Techniques, Wiley Publishing Inc.
10. Mark R. Wilkins & Chris Kazmier (2003), MEL Scripting for Maya Animators, Morgan Kaufmann Publishers
11. Alias Wavefront (2004), Learning Maya 6: MEL Fundamental, Alias Learning Tool.
12. Alias Wavefront (2004), Learning Maya 6: Dynamics, Alias Learning Tool.
13. Alias Wavefront (2004), Learning Maya 6: Expression, Alias Learning Tool.
14. P.A. Egerton and W.S. Hall (1999), Computer Graphics, Mathematics First Steps, Prentice Hall
15. Kian Bee Ng (1999), Digital Effects Animation using Maya, Charles River Media
16. P.Abbott (1997) , Trigonometry, Hodder & Stoughton
17. Th. Driemeyer (1986-2000), Rendering with mental ray, Springer Wien New York
18. Qin Jun,Liao Guang-xuan, Wan yu-tian, Fan Zhi-hang (2002), Experimental Study on Model Fire WhirlWind, Retrieved 10th August,2009, from http://data.fire.net.cn/qikan/UploadFile/200731413426185.pdf

Reference (Website)

1. CGTalk (1997-2006), CGTalk Forum—fluid tornado…, Retrieved 3rd July 2009, from http://forums.cgsociety.org/showthread.php?f=86&t=246694&page=2&pp=15
2. Wikipedia (2009). Vortex, Retrieved 3rd July 2009, from http://en.wikipedia.org/wiki/Vortex
3. Michael McKinley(2009). Michael McKinley (personal page), Retrieved 15th July 2009, from http://www.mtmckinley.net/
4. Morten Moen (2001), Digital Visual Effects and Modelling Animation, Retrieved 15th July 2009,from http://www.morten-moen.net/index.html
5. Autodesk (2009), Maya Online Reference, Retrieved 15th July 2009, from http://download.autodesk.com/us/maya/2009help/index.html?url=Texturing_and_shading_fluids_Shade_particles_using_the_fluid_shape_shader.htm,topicNumber=d0e436338
6. TOI-Pedia (2009), MEL UI Layout, Retrieved 7th July 2009, from http://wiki.bk.tudelft.nl/toi-pedia/MEL_UI_Layouts#rowColumnLayout
7. Paras Chopra (2009), Particles System, Retrieved 13th July 2009, from http://paraschopra.com/tutorials/particle-systems/
8. Alex Alvarez (2009), Candle Flame, Retrieved 20th July 2009, from http://www.thegnomonworkshop.com/tutorials/candle.html
9. VFXTalk (2007), Mental Ray problem (jittering and Alpha holes), Retrieved 20th July 2009, from http://www.vfxtalk.com/forum/mental-ray-problem-jittering-and-t8776.html
10. Tomohiro Abe (2009), MEL Quick Reference, Retrieved 20th July 2009, from http://www.not- enough.org/abe/manual/maya/quick-ref.html
11. Magdalene Chan (2009), Particle Effect for 3D character—Fire Whirl, from http://nccasymposium.bmth.ac.uk/Chan_Magdalene/index.html
12. yahya99 (2009), fume fx - firey whirl test, Retrieved 20th July 2009, from http://www.youtube.com/watch?v=FM9ag6w4QY0
13. AcmeDesign(2007), Fire Tornado, Retrieved 15th June 2009, from http://www.youtube.com/watch?v=t6A6ZS3AHGk
14. Htogrom (2009), Maya 2009 Fluid particle shader, Retrieved 20th July 2009, from http://www.youtube.com/watch?v=FM9ag6w4QY0
15. NetDragon Websoft Inc (2009), Online Game, Eudemons (魔域),Retrieved 20th August 2009,from http://eo.91.com/index/
16. walkalongaviation, (2007), Fire Tornado, Retrieved 10th August,2009, from http://www.instructables.com/id/Camp-Fire-Tornado/

Reference (Video)

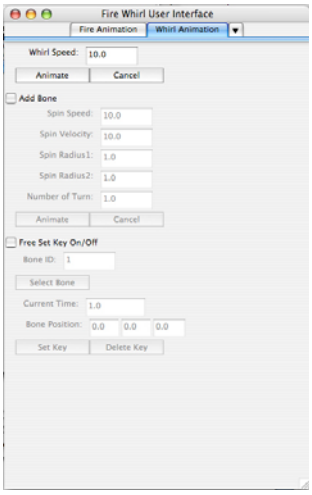
1. Digital Tutors (Director),(2008) , Mental ray Nodes in Maya [Tutorial Video], Oklahoma City: PL Studio Inc.
2. Digital Tutors (Director), (2008), The Artists Guide to MEL[Tutorial Video], Oklahoma City: PL Studio Inc.
3. Digital Tutors (Director), (2008), Mastering Maya Render Nodes [Tutorial Video], Oklahoma City: PL Studio Inc.
4. Digital Tutors (Director), (2008), Fundamentals of Maya Dynamics: Particles and Fields [Tutorial Video], Oklahoma City: PL Studio Inc.
5. Digital Tutors (Director), (2008), Maya Unlimited: Fluid Effects [Tutorial Video], Oklahoma City: PL Studio Inc.
6. David Schoneveld (3D Artist), (2008), Maya Fluid Effects--Pyrotechnics [Tutorial Video] The Gnomon Workshop: Luma Pictures
7. Hajime Yatate (Creator). (1991-1992), The Brave Fighter of Sun Fighbird. Japan: Sunrise

Appendices

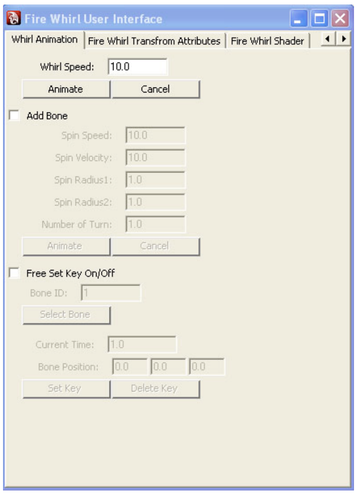
A1 Fire Whirl User Interface Guide

This section will introduce how to use Fire Whirl User Interface to create and animate fire whirl.

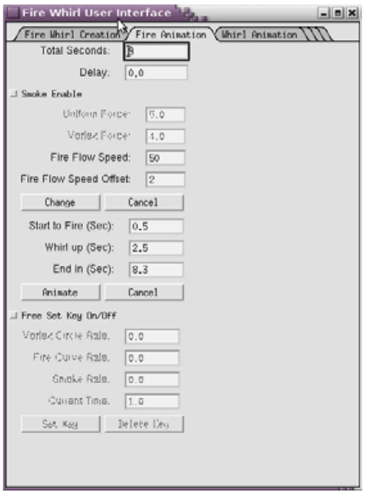
This is the user interface:



Mac Version



Window Version



Linux Version

This guide will explain the usage using Mac UI version.

1. **Fire Whirl Creation**

   The user has to go through this tab first, because this tab contains the controls for creating the fire whirl.

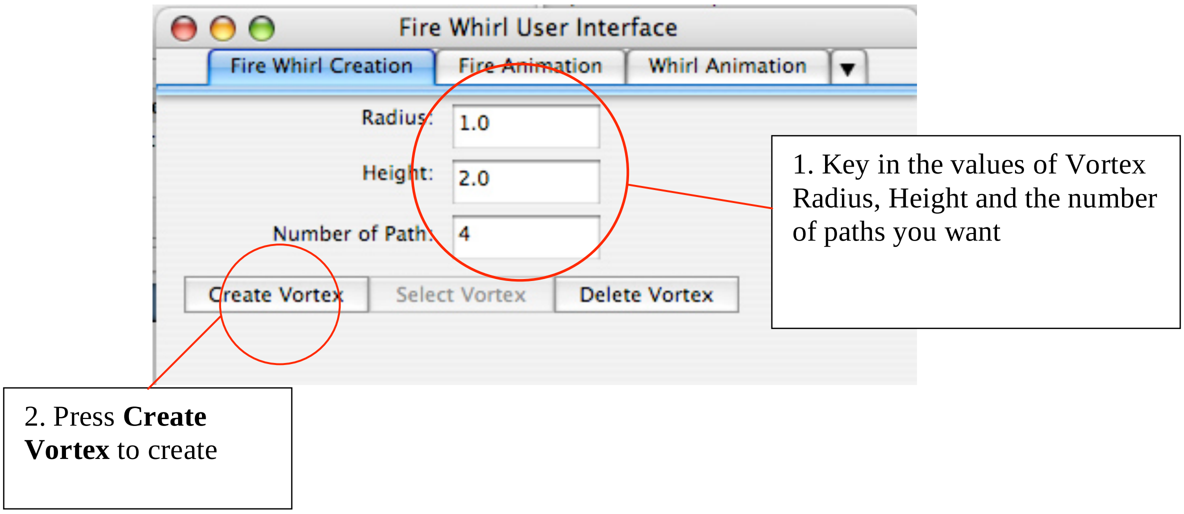
   Radius—the radius of the vortex, it has to be defined, so that the curve path can be calculated (the detailed calculation can be seen in the Thesis. The radius had been set between 0.1 (min number) to 10 (max number)

   Height—this is the height for the vortex curve path. Its minimum height is 1.

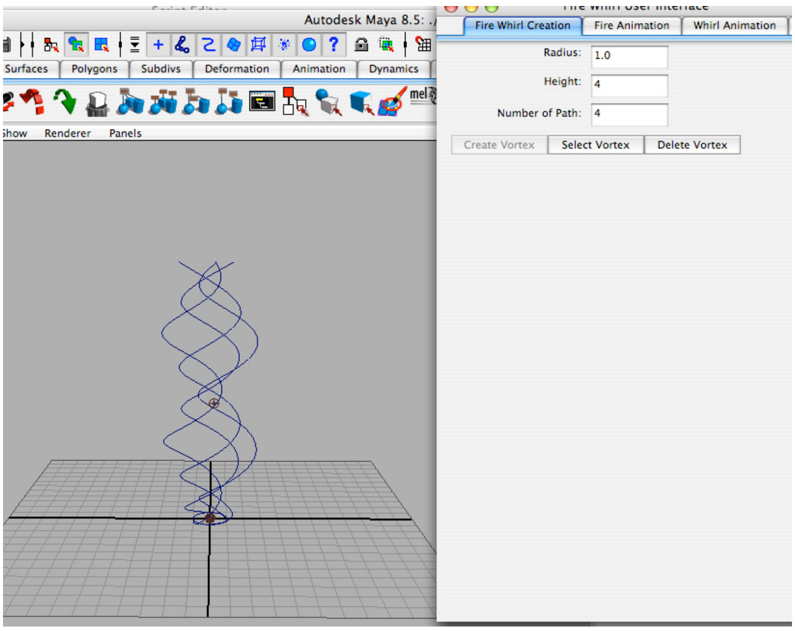
   The user can draw as tall as he wants. However, please note that the taller the fire whirl, the slower the particles animation as the particles takes longer time to finish the path.

   Number of Path—default is 4. According to the real fire whirl seen in you tube, the fire whirl is formed by 4 vortex paths normally. The user can input higher paths, up to 10 paths. (but it is not encourage to set the path height more than 5, it will slow down the particles animation and rendering.

   Fire Whirl Creation interface and the steps of using it are in the following:



The Vortex will be produced as follow:



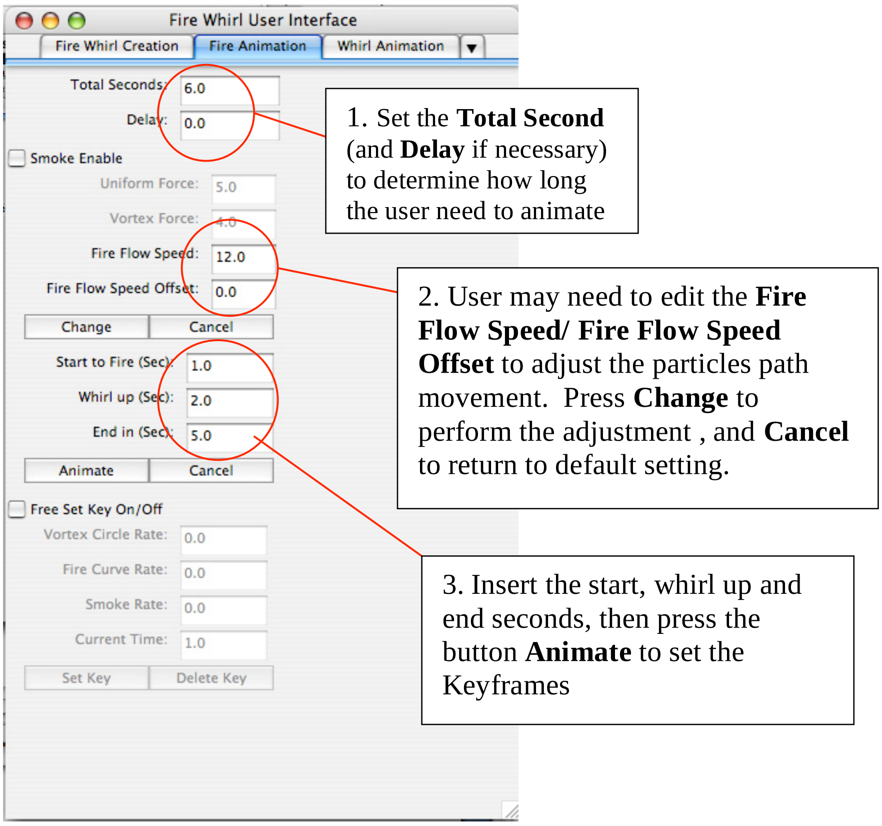
Create Vortex only executes once. When the fire whirl is appeared in the scene, this button will be disable. On the other hand, **Select Vortex** button will be enable for choosing the fire whirl model. To make it enable again, press **Delete Vortex. Create Vortex** button will be enabled again once the fire whirl model is disappeared.

Note for Delete Vortex: this button is executable even the bone, smoke and shaders are presented.

2. Fire Animation

User can choose to animate the fire or whirl after the fire whirl is created.

This is the Fire Animation interface and the steps of using it:



Total Seconds—referring to how long the animation last, when the user keys in 6 Seconds, the total playback frames will be 24 x6, 144 frames).

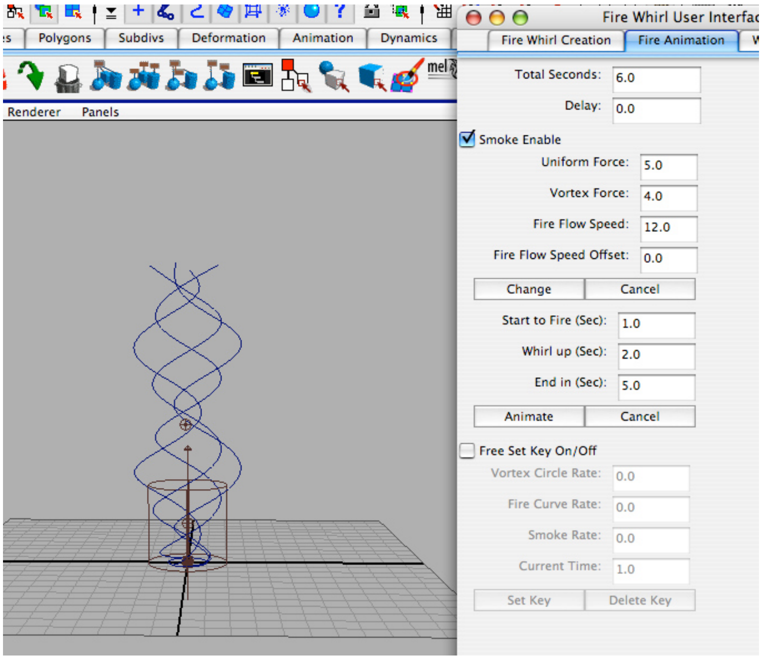
Delay—In case of the user needs longer total playback frames, this field is a option for him to lengthen the total playback time. E.g.

Total Second: 6

Delay: 1

It will result 7 x 24=168 frames in playback display.

Smoke Enable—User can choose if he would like to have smoke in the fire whirl animation. If this is clicked, the scene will be like this:

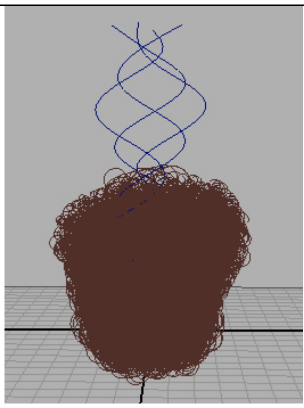
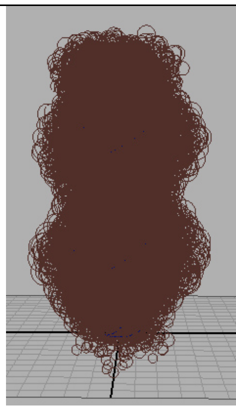


the smoke is created with the cylindrical emitter. The Uniform and Vortex Force field are available to alter.

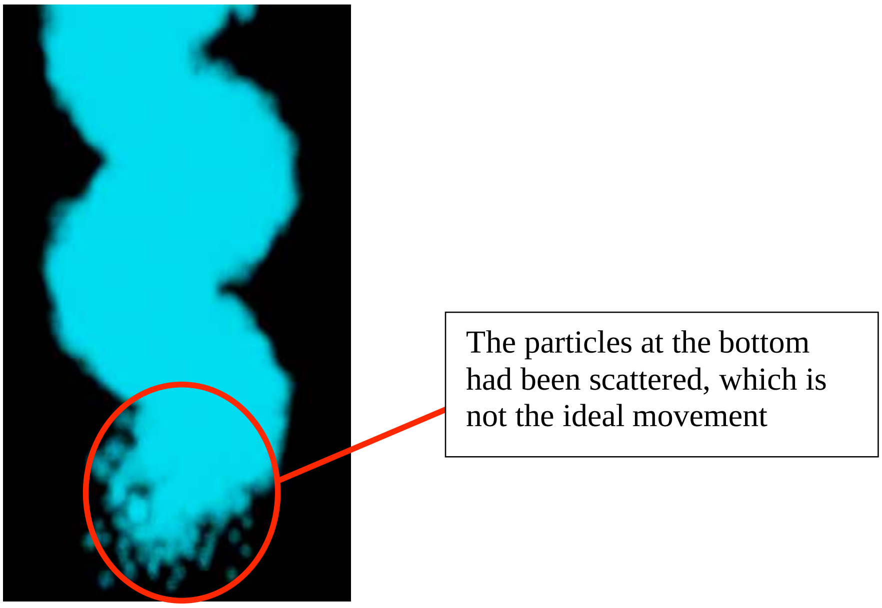
Uniform Force—This is the uniform force control for the smoke. This field is only available when the smoke is created. It adjusts the Uniform Force magnitude.

Vortex Force—it is the same as uniform force control, which is only allowed to adjust when the smoke is presented. It adjusts the Vortex Force magnitude.

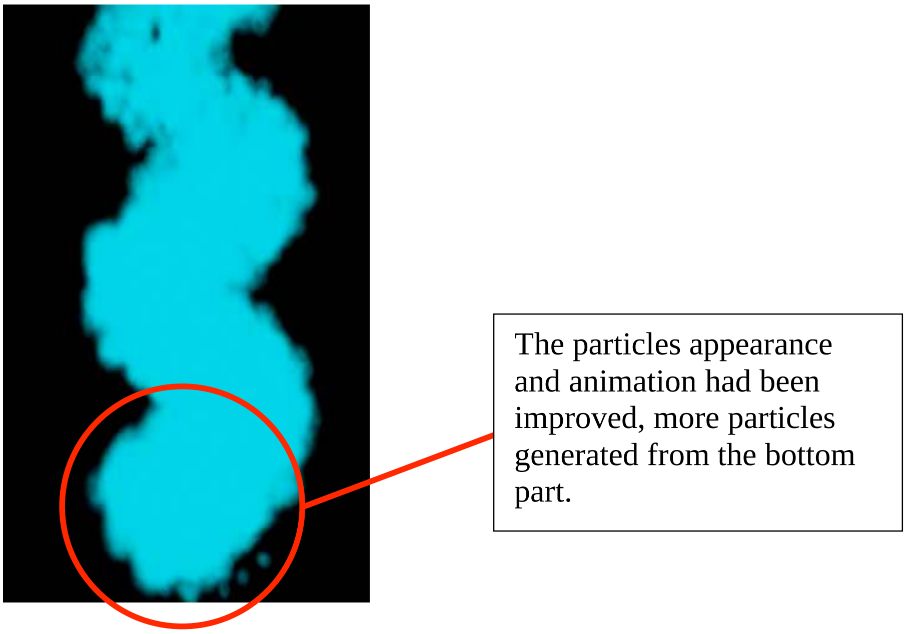
Fire Flow Speed—it is used to adjust the speed and the spread of the fire flow. The default speed is 12, which is not enough to make the particles flow through the whole paths. The following table shows the different Fire Flow Speed in the same frame (frame 28 in both case).

	
Fire Flow Speed in default setting (12)	Fire Flow Speed in 50

Fire Flow Speed Offset—this field allows user to shift the particles closer to the vortex circle particles. When the fire whirl is created in higher path height, the particles will be offset easily. The following images shows how the Speed Offset helps to improve the particles movement:



By inputting 6 in the **Fire Flow Speed Offset** field and press **Change:**



Start to Fire (Sec)—User can define what time the particles starts to generate the particles (vortex circle).

Whirl Up (Sec)—User can also define the time when the fire path particles starts to go through the paths

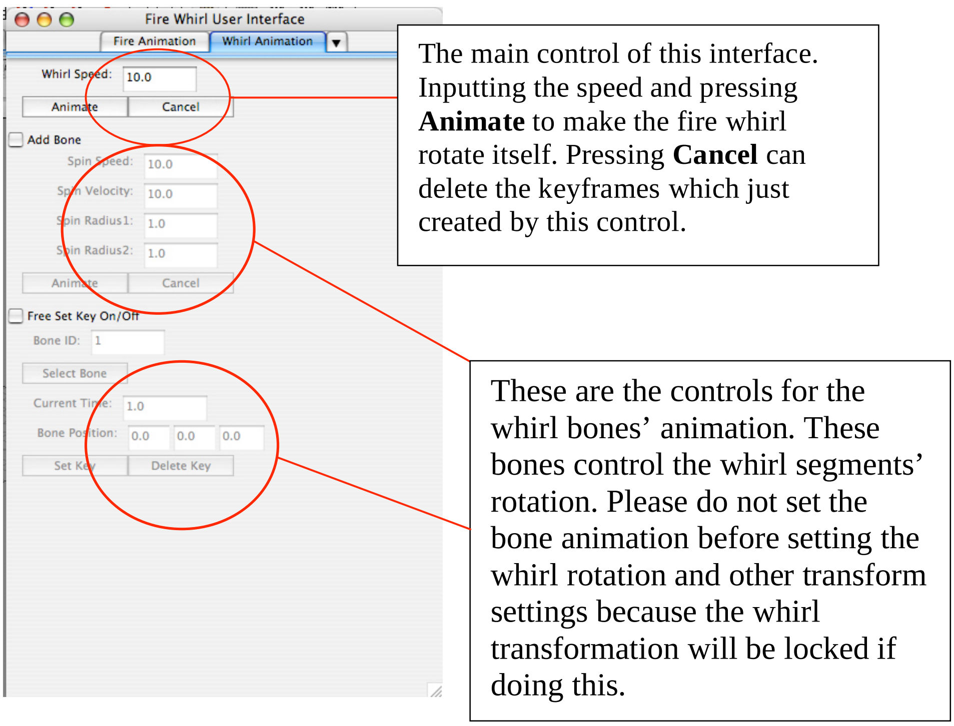
End in (Sec)—similarly, user can define when the particles stops animate.

Once the values are inserted, press **Animate** to perform the keyframes setting. The user can delete the keyframe setting by pressing **Cancel**.

The bottom section of Fire Animation interface is **Free Set Key option**. It provides the controls of Smoke, Vortex circle and Vortex path rate changes and **Current Time** setting. The user keys in the time he want to go to, adjust the rates and press **Set Key** to create a Keyframe. **Delete Key** is given for deleting a single Keyframe in any frame in the timeline. By default, this option is disable, clicking the check box switch can activate it.

3. Whirl Animation

This Interface mainly deals with whirl animation (rotation). The following image is the interface and the steps of using it:



Whirl Speed—control the whole fire whirl rotation. Its calculation is related to the total second. The Keyframes are allocated according to this speed control and the total seconds. Hence, it is advisable to set the total second first then input and perform this control.

Add Bone—Clicking this can activate the controls underneath (excluding the free set key section) , create the bone and attach it to the whirl. It overrides the transformation, so the normal transformation controls such as position, rotation and scale are unavailable. The whirl rotation and Transform Attribute's control will not work properly. Unclicking this option again can reactivate the normal transformation controls by removing bone from the scene. Bone set up here is used to control the rotation of each segment in the whirl.

Spin Speed—it is the control for spin speed for each bone, according to the angle division and Spin velocity.

Spin Velocity—it decides how fast the segments rotate.

Spin Radius 1—it controls the size of rotations, i.e. how long the distances it spins away from the centre. Radius 1 is the radius control for even bone names.

Spin Radius 2—the radius control for the odd bones. It has the similar function to

The last section for this interface is the same as fire animation—free set key. If the user does not like the system animation setting, or they want to edit from system animation setting, the section can help to do that.

Bone ID—use to select individual bone, e.g. ID:1 >FJ1 (bone name) ID:2 >FJ2 etc. Press **Select Bone** to select it.

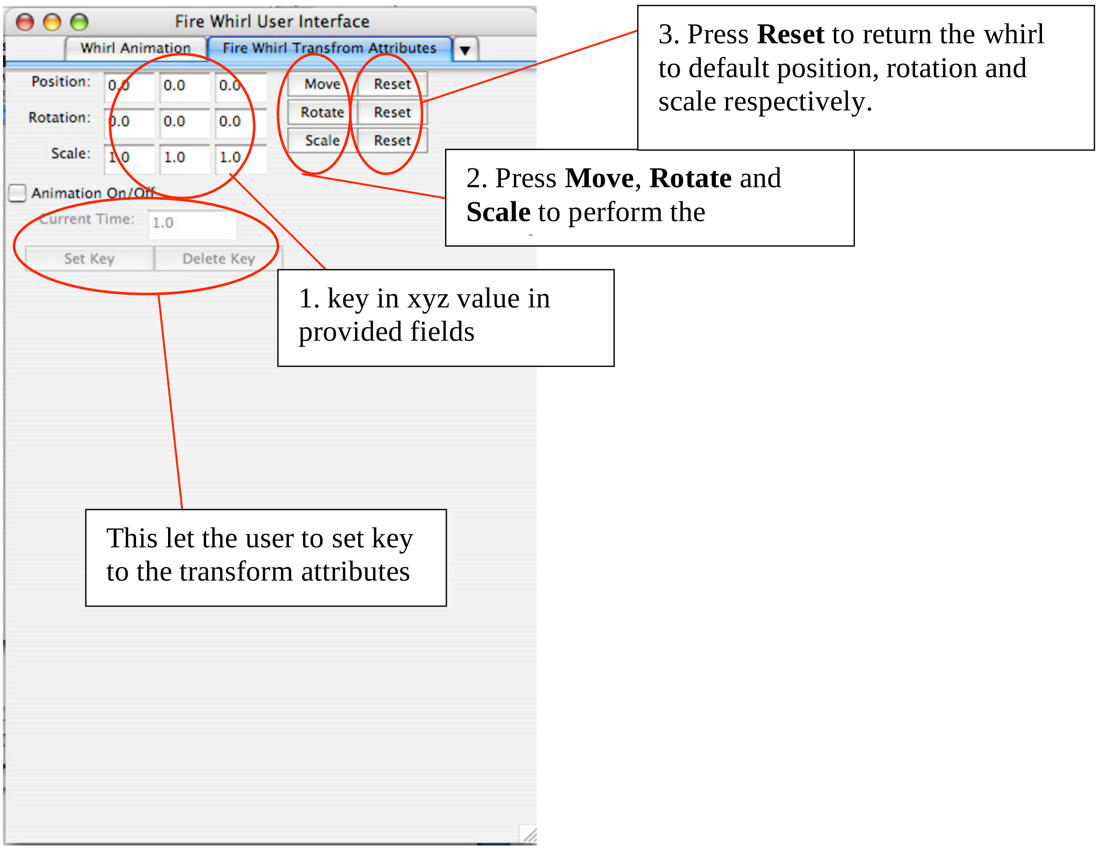
Current Time—set the time to set keyframe

Bone Position XYZ—with setting ID and the position, the user can set translation keyframe to the particular bone. Press Set Key to execute the action. Delete Key is used for delete a keyframe in the chosen time in the timeline.

4. **Transform Attribute**

This is the general transformation control for fire whirl. This interface allows the user to move, rotate and scale the whole fire whirl.

This is the interface:

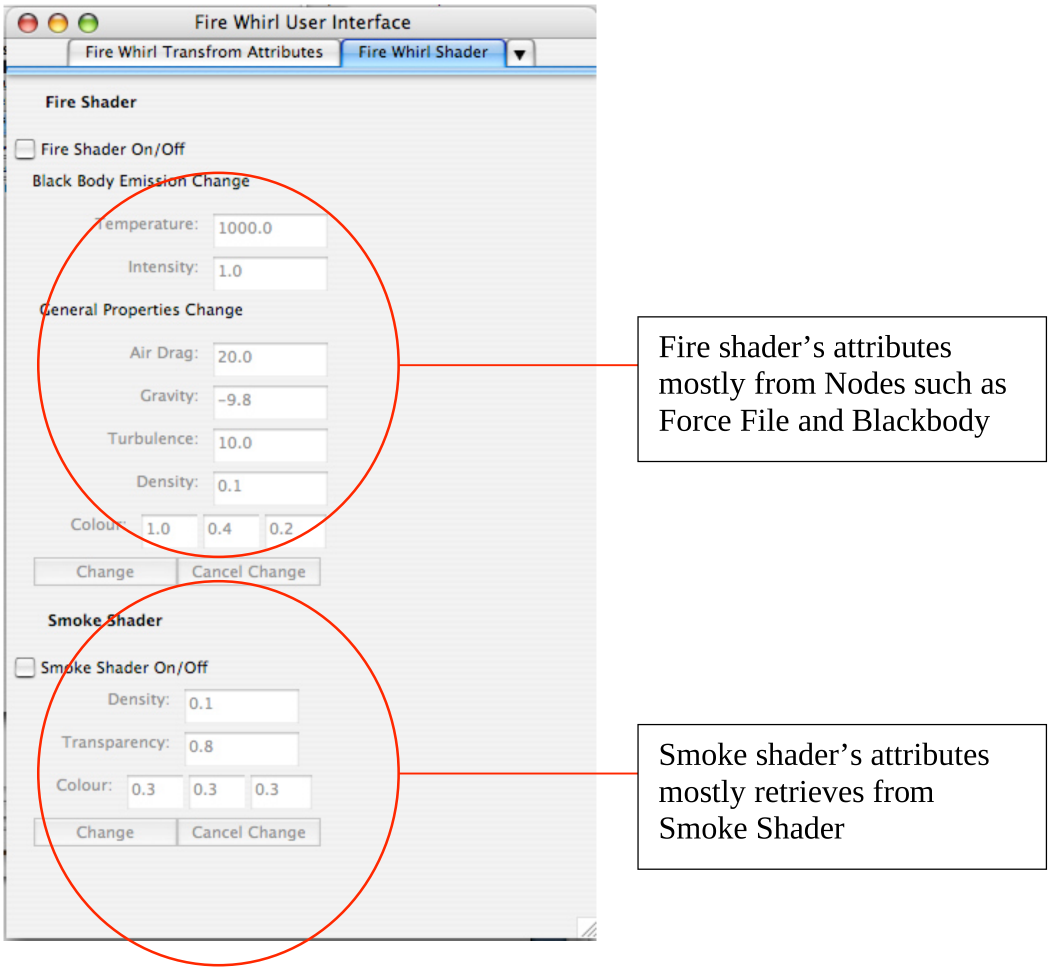


It can set keyframe(s) freely one by one like Fire and Whirl Animation, click the **Animation On/Off** to actuvate the controls.

Note to these Attribute controls—if the user want to animate the general transformation, it is strongly recommended to animate this first before all bones, whirl and fire path, in order to get the accurate animation for animating transformation, bones, whirl and fire path al together.

5. **Fire Whirl Shader**

This interface is classified into two sections—Fire Shader and Smoke Shader.



Fire Shader section

Temperature—this control is the field for Mental Ray Black Body temperature adjustment. It is calculated as Kelvin.

Intensity—it is the control for Mental Ray Black Body intensity adjustment.

Air Drag—it is used for adjust Drag force magnitude.

Gravity—it is used for adjust Gravity force magnitude.

Turbulence—it is used for adjust Turbulence force magnitude.

Density—it is the control for changing Fire shader density value.

Colour (RGB—it is the colour change of Fire Fluid Incandescence colour. It can change the fire colour from there.

These controls are activated only when the fire shader is on. The values of provided attributes can be changed by pressing **Change** button. **Cancel Change** button returns the values to default settings.

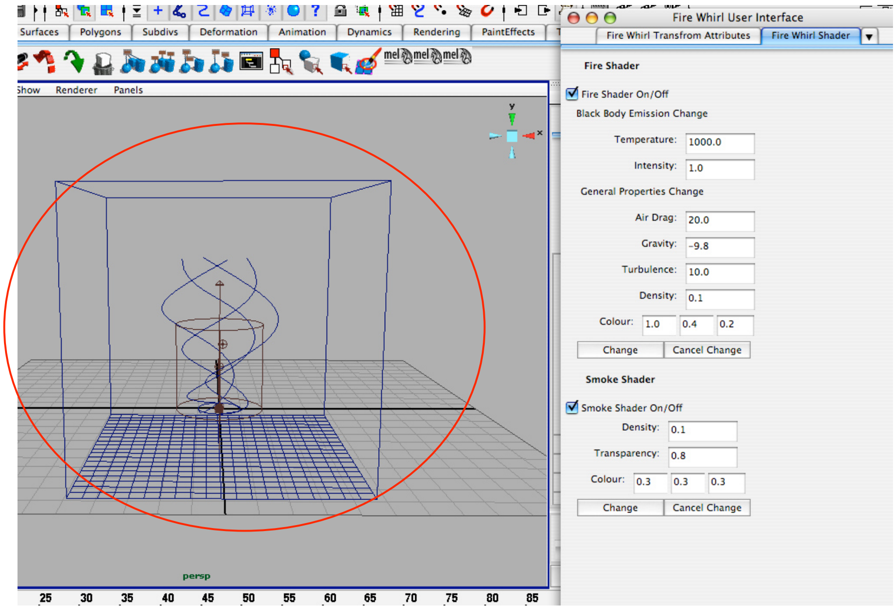
Smoke Shader section

Density—the value change for Fire Shader density attribute.

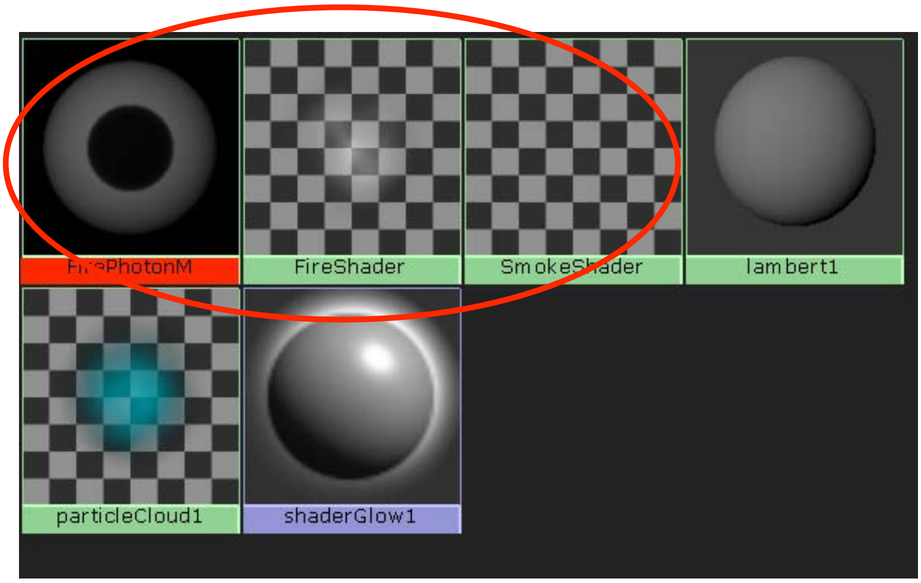
Transparency—the value change for Fire Shader Transparency.

Colour (RGB)—the colour change for Fire Shader colour

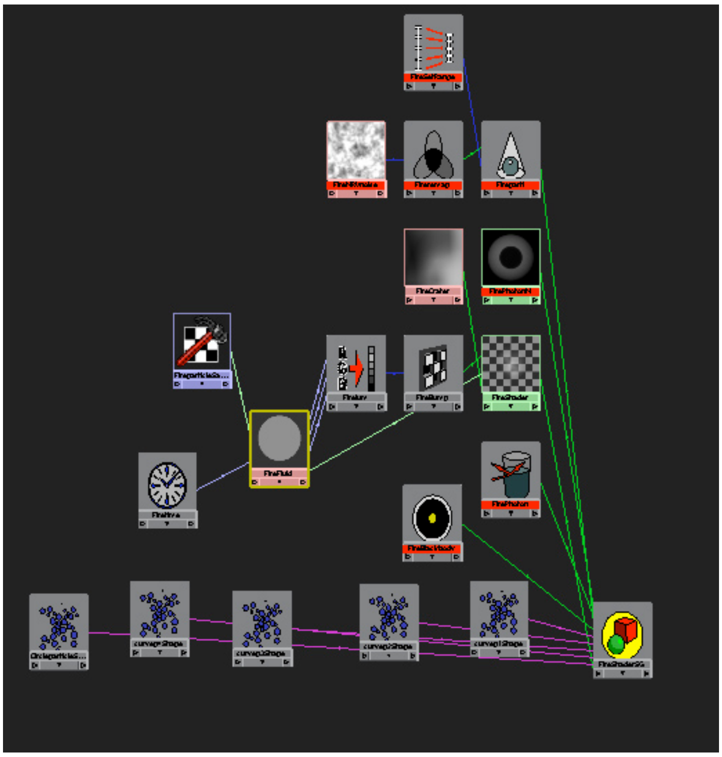
The following pictures are the result of switching on both fire and smoke (provided that the smoke is presented in the scene) shader.



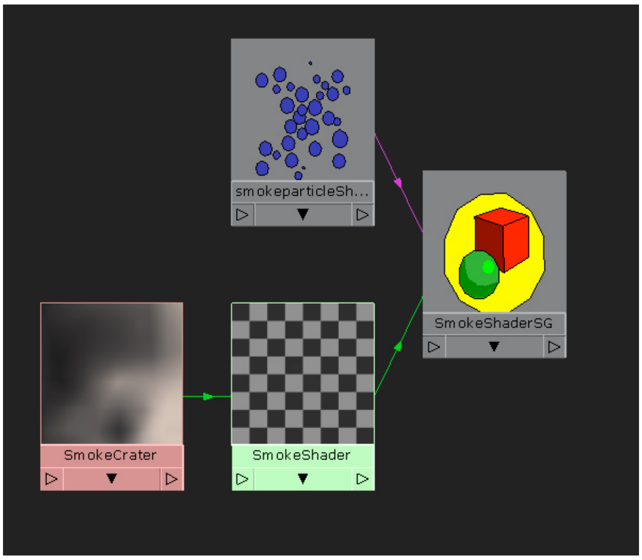
The Fluid container will be seen with assigned position in the scene, if the shader is successfully executed.



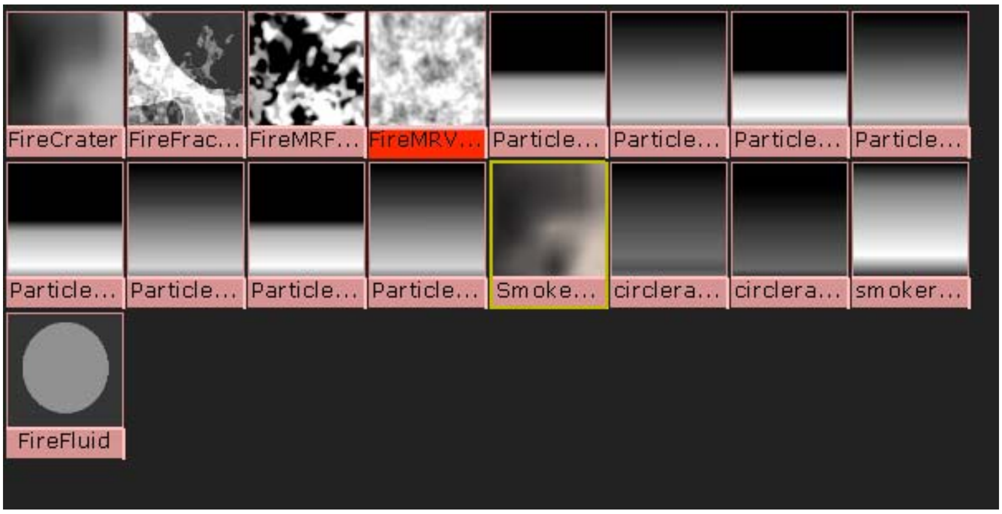
The circled shaders are the added shaders from the MEL script-- mental ray photon Shader and Particles Cloud Shader.



This is the Shading Newtork of the Fire Shader.



This is Shading Network of smoke Shader.

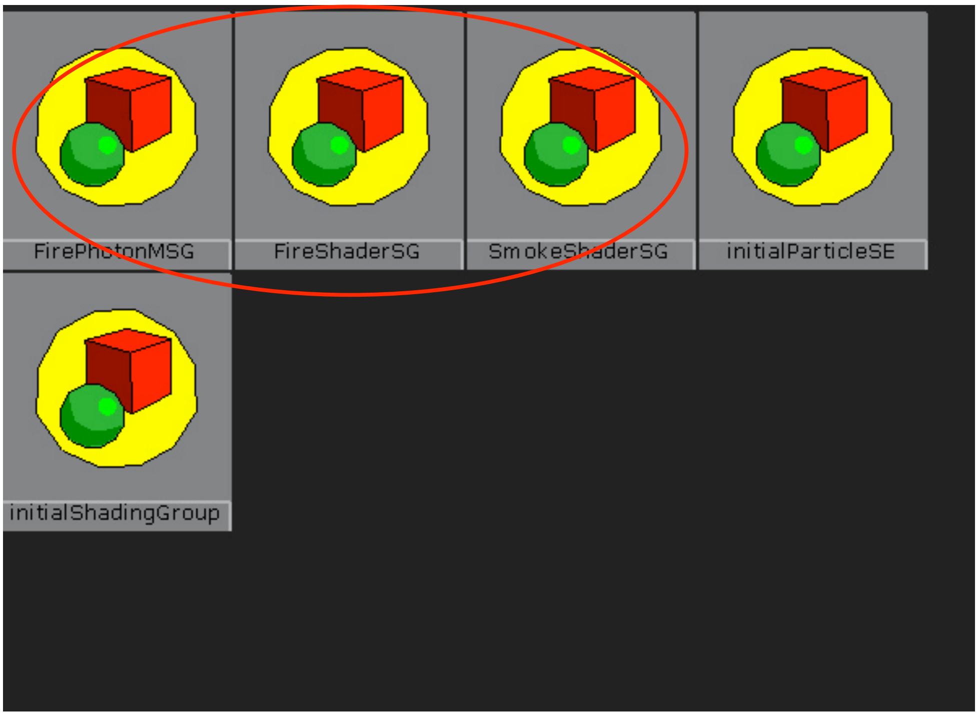


The texture network, it includes all the textures and ramp maps used in this fire whirl. Textures start from the top left:

Crater (fire), Fractal (fire), Volume Noise (fire mental ray), Fractal (fire mental ray), radiusPP and opacityPP ramps(Fire Path Particles), Crater (Smoke), radiusPP and opacityPP ramps (Vortex circle Particles), radiusPP ramp for smoke, Fire Fluid texture

All the utility node are related to Fire Shader. Starting from the left:

Black Body, Bump Map, Dielectric material, Set Range, Luminance, Parti Volume Photon Shader, Particles Sampler Info, Remap Values, Time.



Three Shading Engines (Photon, Fire and Smoke) are supposed to create when shaders are on.

****

If the fire whirl is closed halfway of fire whirl creation, the user can reopen it and use the animation sections to make the animation, or deleting the existing fire whirl by typing the data in Fire Whirl Creation section, then create it and redo again.

This Fire Whirl User interface can only create **one** fire whirl.

A2. Equations Related to the Code

1. **Angular Velocity**

   $\omega = d\theta/dt$

   ω is Angular Velocity

2. **Particle Goal Speed Formula**

   Goal U=age*Fire Flow Speed-Fire Flow Speed Offset

3. **Incompressiable Flow Fluid Dynamics Equation (Navier-Stokes)**

$$\frac{V^2}{2} + G + \frac{\text{pressure}}{\text{density}}$$

where G is gravity, V is fluid flow speed

$$\overbrace{\rho\Big(\underbrace{\frac{\partial \mathbf{v}}{\partial t}}_{\text{Unsteady acceleration}} + \underbrace{\mathbf{v}\cdot\nabla\mathbf{v}}_{\text{Convective acceleration}}\Big)}^{\text{Inertia (per volume)}} = \overbrace{\underbrace{-\nabla p}_{\text{Pressure gradient}} + \underbrace{\mu\nabla^2\mathbf{v}}_{\text{Viscosity}}}^{\text{Divergence of stress}} + \underbrace{\mathbf{f}}_{\text{Other body forces}}.$$

4. **Circle Formula**

   X=Rsin(θ)

   Y=Rcos(θ)

5. **Helix Formula**

   $r(\theta) = R\cos(\theta)\,\mathbf{i} + R\sin(\theta)\,\mathbf{j} + C\theta\,\mathbf{k}$

6. **Newton Second Law**

   F=ma

7. **Dnoise Function**

   (dnoise(position x 0.5 x scale+(-3) x time)+0.9) x lifespan

8. **Sphere Random Function**

   *sphrand(VR*2)*

   where VR: Vortex Radius

9. **line Step and Smooth Step function**

   linstep(0, lifespanPP, age)

   smoothstep (0, lifespanPP, age)

   0 is the start point, lifespanPP is the end point and age is the parameter.

10. **Total frame calculation (in the code only)**

    $finaltotalframe=$totalframe+$delayframe;

    *where $totalframe=$animatetime*24;*

    *$delayframe=$animationdelay*24;*

11. Speed- time or Relativity Related Equation (in the code only)

$KeyframeGap=$totalframe/$VortexSpeed

$KeyframeGap is the average time, $totalframe is the Total frame in the total playback frame and $VortexSpeed is the speed

(in bone animation)

$angulardiv=$totalframe/$SpinSpeed/$SpinVel

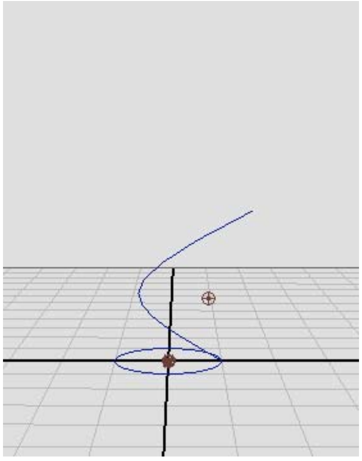
Difference on Spin Speed and Spin velocity in the script:

Spin Speed is the speed of each bones

Spin Velocity is the average speed for all the bones.

A3. Fire Whirl Model

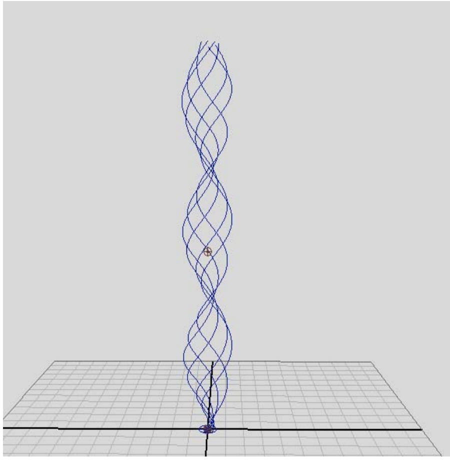
This section will show the fire whirl model in different height, radius and number of path in 5 images.



Radius:1

Height:1

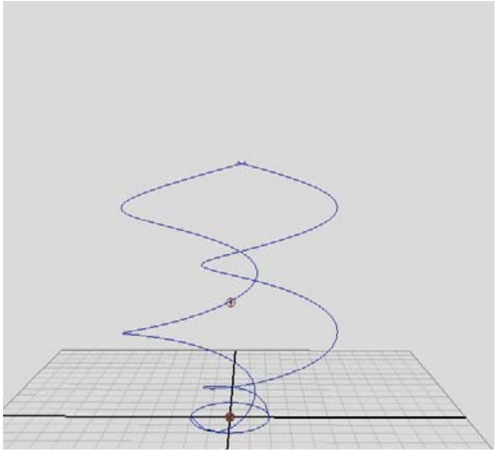
Number of Path : 1



Radius: 0.5

Height: 5

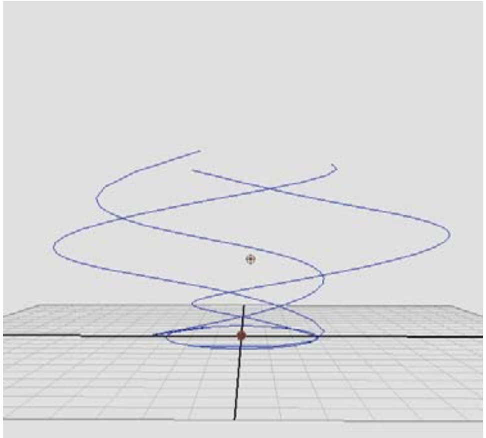
Number of Path: 6



Radius: 2

Height: 3.5

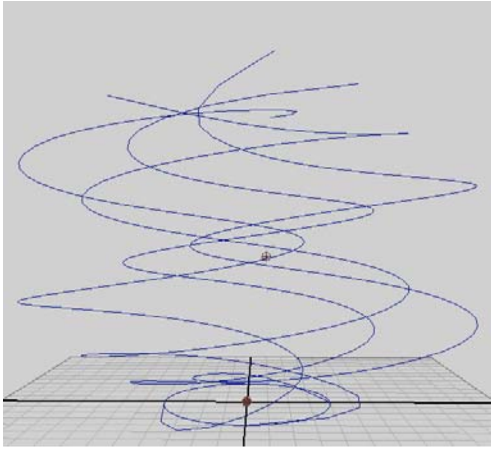
Number of Path: 2



Radius:3

Height:2

Number of Path : 3



Radius:4

Height:4

Number of Path : 4

A4 Schedule Diagram

This is the schedule diagram mentioned in Section 1.4