

FIREFLY MANUAL



Index

- [Object Descriptions](#)
- [Basic Node Properties, Actions, Conditions & Expressions](#)
- [Firefly Engine Object](#)
- [Camera Object](#)
- [2D Text Object](#)
- [Image Object](#)
- [Material Cache Object](#)
- [Firefly Movement](#)
- [Animated Mesh Object](#)
- [Firefly Node - Particle System](#)

Object Descriptions



Firefly Engine

This Firefly Engine object does almost all the work within these set of extensions. First, you should add one to the frame and then resize it to cover the frame. The area covered by this extension dictates the resolution of your 3D scene.



Firefly Node Camera

The Firefly Node Camera object is necessary to draw anything to the frame. It works similarly to how the Engine object works, except the area covered by the Camera object dictates where to draw the scene to (through the camera). Through this system, split-screen effects can be achieved, but usually the Engine object and your singular Camera will both be at the same point on the frame and both be of the same size.



Firefly 2D Text

The Firefly 2D Text object puts 2d text inside your 3d world useful for things like names above heads and other User Interface information.



Firefly Image

The Firefly Image object puts images inside your 3d world useful for backdrops for all that User Interface information you have.



Firefly Material Cache

The Firefly Material Cache object represents and acts as an array of Materials. A Material has an Index that it resides at within a Material Cache object, and it can also have a name. Most node objects require one or more materials to render something other than a grey mass in the scene.



Firefly Movement

The Firefly Movement object is one of the most important objects in the FireFly set as it helps you move all your other nodes around.



Firefly Node Animated Mesh

With the Firefly Node Animated Mesh node you can insert an animated mesh into the scene. An Animated Mesh is animated across a number of Frames by calculating the position, scale, and rotation of the mesh's Joints using each one's collection of Keyframes across those Frames.

So, an Animated Mesh might have many Joints, and each Joint can have many Key Frames of a Position and/or Scale and/or Rotation type. When the animation takes place, it calculates what Position, Scale, and Rotation each Joint should be set to at a given frame by working out each difference between the last and next Key Frame.



Firefly Node Billboard

The Firefly Node Billboard object is like a 3d sprite: A 2d element, which always looks to the camera. It is usually used for explosions, fire, lens flares, particles and things like that.



Firefly Node Dummy

The Firefly Node Dummy object isn't really an object at all, in fact it isn't anything and you will never see it in through the Node Camera but it's incredibly useful for holding data for example you could have players mark an area in your world and put a dummy node there and use its values in your calculations.



Firefly Node Light

The Firefly Node Light object places various light sources such as spot lights or directional lights in your scene, Lights are often needed to brighten up a scene. Any Material (from the Material Cache object) that has its 'Lighting' setting set to true/on (which it is by default) means that it requires light to be rendered in some colour other than black.



Firefly Node Particle System

With the Firefly Node Particle System you can insert a particle emitter into the scene. A Particle System node can be one of these types: Point, Box, Sphere, Animated Mesh Node, Cylinder, Mesh, Ring. These types describe the shape of the emitter. What is actually emitted is similar to a Billboard node object.

A Particle System Node also can one or more Effectors. An Affector is something located at a X,Y,Z position within the scene and will affect particles generated by the node object in different ways. The different types of Effectors are 'Attraction', 'Fade-Out', 'Gravity', 'Rotation', 'Scale'.



Firefly Node Primitive

The Firefly Node Primitive object is capable of very quickly creating a mesh in the shape of standard objects such as cubes, spheres, arrows, tubes cones, cylinders and planes.



Firefly Node Static Mesh

The Firefly Node Static Mesh object isn't static as in not able to move its just static as in not containing any animations, most of your 3d objects will be static meshes.



Firefly Node Terrain

The Firefly Node Terrain object is capable of very quickly loading terrains and updating the indices at runtime to enable viewing very large terrains. It uses a CLOD (Continuous Level of Detail) algorithm which updates the indices for each patch based on a LOD (Level of Detail) which is determined based on a patch's distance from the camera.



The Firefly Node Text object lets you insert 2D text within the 3D scene. It works similar to the Billboard object in that it will always face the camera and the size of the text as it appears is not dependant on how close the object is to the camera.

Firefly Node Text



This Firefly Node Water object for rendering an animated water surface mesh. Looks really good when a Material with Transparent Reflection is used.

Firefly Node Water



Firefly Skybox

This Firefly Sky object object is not quite a node object in that you can only have one in the scene, and it cannot be moved around. you can however choose to have a skysphere or a skybox which will allow your players to feel that the world is massive around them.

Basic Node Properties, Actions, Conditions & Expressions

All the 'node' objects share these actions/conditions/expressions:

Actions

- Add to Firefly Engine
 - Select an object - This action adds the object to the Firefly Engine object instance specified in the parameter. This is necessary so that the engine knows about the Node object.
- Remove from Firefly Engine
 - Select an object - This action removes the object from the Firefly Engine object instance specified in the parameter.
- Node Properties
 - Set Parent Node
 - Enter Fixed Value of an Firefly Node (0 for none) - This action sets another Firefly – Node object (any type of Node) as the parent of this object. Which then means its Position, Rotation, Scale, etc settings are relative to its parent.
 - Position, Rotation, Scale
 - Position(X, Y or Z) - Move the node along either the X, Y or Z planes using the expression editor.
 - Rotation(X, Y or Z) - Rotate the node along the X, Y or Z axes
 - Scale(X, Y or Z) - Scale the node in either of the 3 dimensions
 - , Visibility
 - This boolean setting denotes whether the object is visible in the scene or not.
 - Automatic Culling - Determines if objects outside of camera view are drawn or not
 - Box - Camera bounding box draws invisible lines to the object bounding box, if any lines connect then the object is drawn.
 - Frustum Box - Draws an invisible bounding Box and checks if the camera can see any part of it.

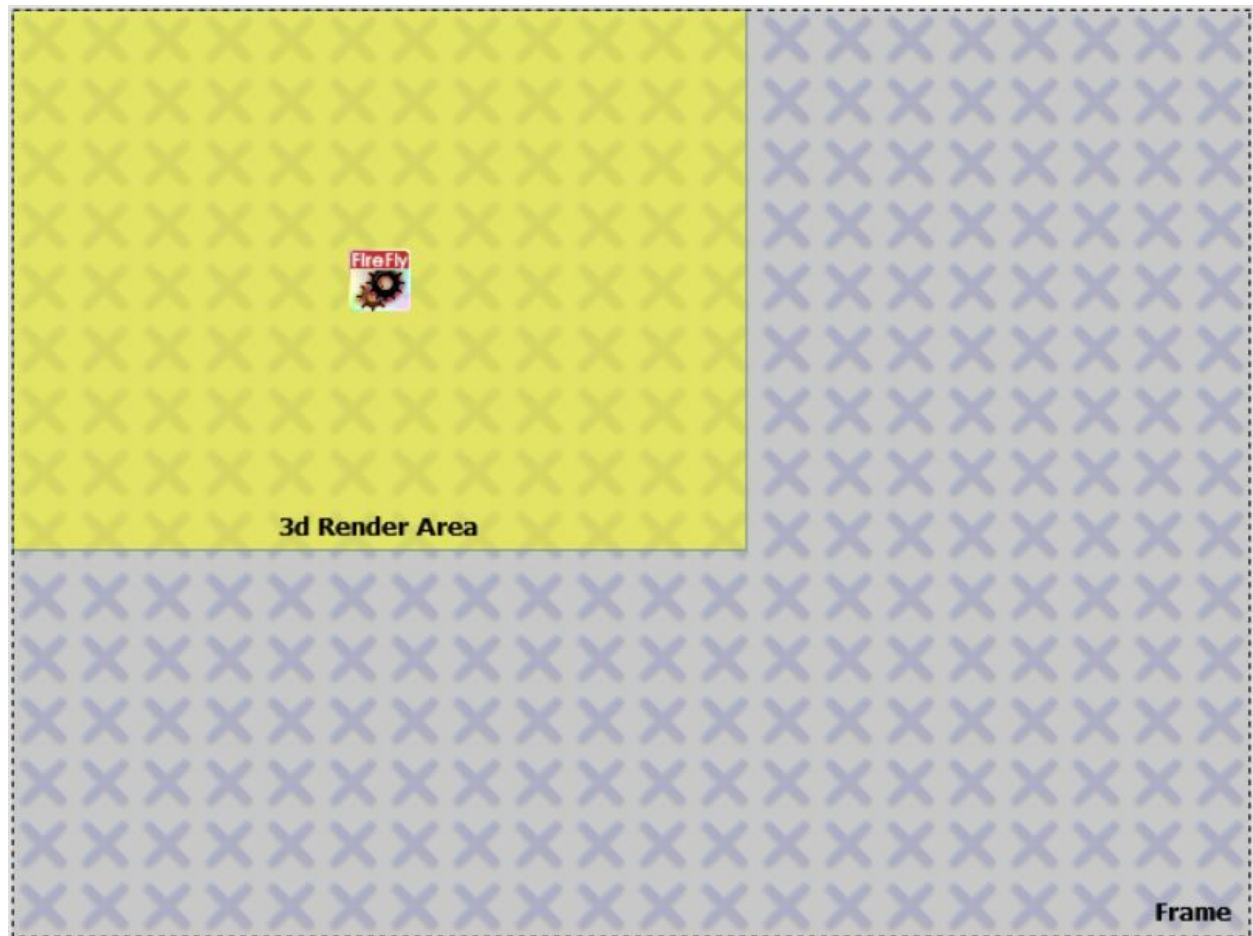
- Frustum Sphere - Draws an invisible bounding sphere and checks if the camera can see any part of it
 - Off - Never hides meshes out of sight, this can have a huge negative impact on larger scenes.
 - Set - Allows you to set which mode the Automatic culling uses during runtime.
- Draw Debug Data
 - Off - Turns off debug data
 - On - Turns on debug data
 - Set - Sets if debug data like bounding boxes should be drawn (0 off 1 on)
- Collisions - Collisions work like this: When node A (that Uses Collisions) might collide with node B (that Uses Collisions, it compares node A's Collision Radius and Translation against node B's mesh. If they collide, it is that node A has collided with node B, not the other way around. If node A's collision Radius is equal to 0, then it cannot collide with node B. For node B to collide with A, it has to compare B's Radius and Translation against A's mesh.
 - Use Collisions - If this is set true, then this Node can collide and be collided into by other colliding nodes.
 - Radius - If this node Uses Collisions, then this is the radius of the space that it takes up around its position when it collides into Meshes of other nodes (that use collisions also).
 - Translation - If this node Uses Collisions, then this is an offset from its position that it uses to calculate when it collides into Meshes of other nodes (that use collisions also).

Conditions

- Is added to Firefly Engine - Has this object been added to an Firefly Engine instance?
- Node Properties
 - Has Parent Node - Has this object been set a parent node?
 - Position > Compare (X,Y or Z) of a Position - Using the expression editor compare on the three three dimensional position values to a expression.
 - Visibility
 - Is visible - Is the object set to be visible or not?
 - Is truly visible - Will return true if the object and all of its possible parent nodes are visible also.
 - Is Drawing Debug Data - Returns whether Debug Data is set to draw or not.
- Uses collisions - Returns whether Use Collisions has been turned on or not.
- Has collided with a node - Is a true condition that will happen when this node object's Collision
- Radius collides with another node object's mesh.
- Has collided with a particular node - Enter Fixed Value of Firefly Node that the object may collide with Is a true condition that will happen when this node object's Collision Radius collides with another node object's mesh that has the Fixed Value equal to the inputted parameter.

Firefly Engine Object

Firefly Engine object is the base of the entire Firefly system. It is the object that actually takes all of your various nodes and other Firefly objects and renders the 3d scene to the frame. You must have at least one in your projects frame in order for anything 3d to appear when you run it. The size of the object “drag space” determines where in the frame the 3d scene is rendered.

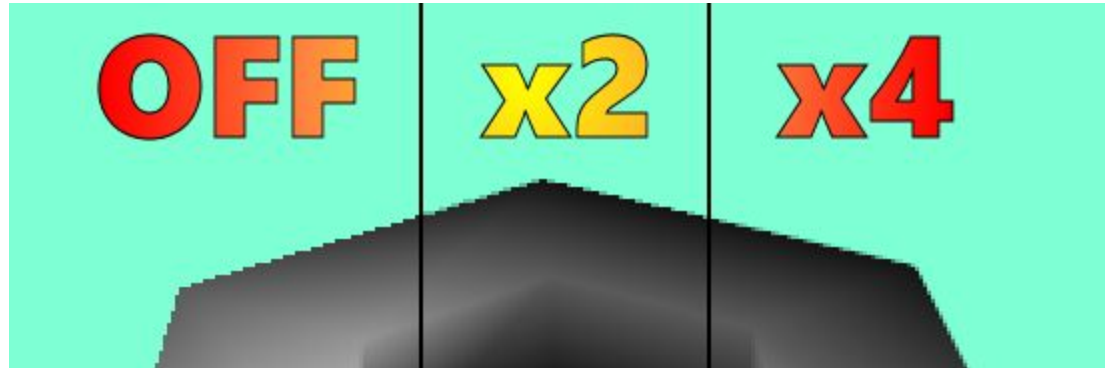


Take a look at some of its specialized functions

Actions

- Start Engine - Allows the engine to start rendering the 3d scene to the frame
- Stop Engine - Stops the engine from rendering the 3d scene to the frame
- Set Engine Resolution - You can with this Action change at runtime the Engine's render space in the frame, but passing two values one for width and another for height in the expression editor.
- Setup Settings
 - Device Driver - Using this action you can select at runtime the driver used by firefly to render the 3d scene, options like OpenGL and Direct3d are here.
 - Anti Alias - antialiasing is a software technique for diminishing jaggies - stairstep-like lines that should be smooth. Jaggies occur because the the monitor doesn't have a high enough resolution to represent a smooth line. Antialiasing reduces the prominence of jaggies by surrounding the stairsteps with intermediate shades of colour. Although this reduces the

jagged appearance of the lines, it also makes them blurry. You can decide whether the effect is off x2, x4, x6 or x8.



- Set Background Colour - Select the empty space background color used by the engine
- Set Ambient Light Colour - Select the default ambient light colour used in engine
- Set Render Speed - Controls how often the Firefly Engine renders everything. Works the same as Active's animation speed.
- Set Fog Colour - Colour of the rendered fog. Fog is applied to materials with the fog flag on.
- Fog Start Distance - The distance to when fog starts
- Fog End Distance - The distance where fog melds into the background. "100% thick"
- Load Resource Package - Point to a ZIP file that can contain all your model resources like Models, maps or textures. Then from the expression editor when needing to access one of these resource for any of your relevant nodes, simply type the resource name in. EX: "mymodel.obj" NOTE: If you load a model that has textures defined within it, they will automatically be loaded and assigned.
- Collisions
 - Set Travel Distance -
 - Manually Detect Collisions -
 - Scene's Collisions -
- On Draw
 - Render to Texture - Saves the camera's view onto a material's texture.
 - Draw 3D Line - Using the expression editor create a line by feeding 2 sets of 3d coordinates, and selecting a texture from Materials Object
 - Draw 3d Triangle - Using the expression editor create a triangle by feeding 3 sets of 3d coordinates, and selecting a texture from Materials Object
 - Draw 3d Box - Using the expression editor create a box by feeding 2 sets of 3d coordinates representing opposite corners , and selecting a texture from Materials Object
 - Draw Pixel - Draw a single pixel in 2d space by feeding the expression editor X and Y values and select color and alpha channel color.
 - Draw Texture - Draw a material's texture onto the camera's view.

Conditions

- Is Firefly Engine running - Triggers when the engine is running each frame
- Is Checking for Collisions -
- On Pre 3D Scene draw - Triggers before anything has been drawn
- On Post 3D Scene Draw / Pre 2D Scene draw - Triggers after every 3D object has been drawn
- Post 2D Scene draw - Triggers after everything has been drawn

Expressions

- Get Engine Width - Returns the width of the engine
- Get Engine Height - Returns the height of the engine
- Setup Settings:
 - Get Device Driver - Returns the device driver renders the engine.
 - Get Anti Alias Mode - Returns the type of anti aliasing type of the Engine.
- Get Background Color - Returns the color of Engine's background
- Get Ambient Light Color
- Get Render Speed
- Get Fog Color
- Get Fog Start Distance
- Get Fog End Distance
- Override Material
- Get Current Time
- Get Current Camera's Fixed Value
- Collisions
- Screen
- Matrices
- Get last debug message

Camera

The Camera object is your "viewport" into the 3d world you create with Firefly. As with the Engine object, you can drag out the object to fill parts of your frame. The area the object covers will be the camera display surface in the app. This means you can have two cameras displaying at the same time. Useful in main situations. You could even make a app capable of VR with this method.

Here is a list of the unique Actions, Conditions and expressions for Camera Object

Actions

- Set Object Width - Set the horizontal size of the viewspace
- Set Object Height - Set the vertical size of the viewspace
- Viewports -
 - Add to Firefly Engine Stack -
 - Insert into Firefly Engine Stack -
 - Remove from Firefly Engine Stack at -
 - Remove all from Firefly Engine Stack 0
- Target - Actions for pointing the camera
 - Set (X,Y & Z) Target - Three separate actions for each vector, where you pass a value to set its relative target vector.
 - Set Target - Pass using the expression editor values for each of the three vectors (X,Y & Z) to determine target vector.
- Up Vector -
 - Set (X, Y & Z) Vector - Sets the coordinate which the camera sees as the "up" direction which it rotates by. In Local coordinates.
- Set Field of View - Set the size of your "eye" in degrees, bigger values will extend the observable game world but view will look more fish eye'd. Default 90, recommended maximum is 120.

- Set Aspect Ratio - Ratio between the camera's width and height, for best results set this to Camera's Width / Camera's Height.
- Set Near Clipping Plane - Allows you to change how close the faces need to be in order for them to not render.
- Set Far Clipping Plane - Allows you to change how far the faces need to be in order for them to no longer render, bigger values will have a greater impact on performance on large scenes.
- Bind Target and Rotation - Allows you to match the camera to a node's rotation and position if previously have set a parent node.
 - Off/On or Set - Three actions allowing you to manually set Off or On or set using a boolean expression
- Mouse Pick - allows you to "Shoot" from the camera and return the first node the "shot" collides with and return its Fixed Value for selection purposes
 - Check Off/On - Sets Off or On the constant use of the Mouse Pick function
 - Check Once - Runs the check once during main loop

Conditions

- Viewports > Is viewport added to Firefly Engine Stack - Checks to see if in fact the viewport is part of the Firefly Engine Stack, Need to select the proper Fixed value from the Firefly Engine object you are testing against
- Binds Target and Rotations - This condition is triggered TRUE when you have Binding set to on

Expressions

- Viewports > Get indices in Firefly Engine Stack - Returns indices of relative Firefly Engine Stack, requires passing of Fixed Value of Engine object in expression
- Target > Get (X,Y & Z) Target - Three separate expressions returning one of the vectors currently used as a camera target.
- Up Vector > Get (X, Y & Z) Position - Three separate expressions returning one of the vectors currently used as a up vector value.
- Get Field of View
- Get Aspect Ratio
- Get Near Clipping Plane
- Get Far Clipping Plane
- Mouse Pick
 - Get Fixed value of Mouse Pick Node - Returns fixed value of node selected during mouse pick
 - Get name of Mouse Pick Node - Returns name of node selected during mouse pick
 - Get (X, Y & Z) Intersection Point from Mouse Pick - Three expressions that return the vector value from the intersection point from the selected vector.

2D Text

Displays a 2d text string above the 3d world in the Firefly engine. Its position is determined just like a string object would be in your 2d project. It does not move with the rest of the 3d scene and always remains on top.

Actions

- Text

- Set font name - Select a new font by passing its name in the expression editor
- Set Size - Set the size of the font
- Set bold - Set the use of bold, boolean 0=off, 1=on
- Set italic - Set the use of italic, boolean 0=off, 1=on
- Set underline - Set the use of underline, boolean 0=off, 1=on
- Set strikeout - Set the use of strikeout, boolean 0=off, 1=on
- Set font color - using a color set expression, change color of font
- Set Text - Pass using the expression editor the contents of the displayed string
- Set Transparency - Determine how opaque with the expression editor the display string is
- Set Object Width - Determine the string display surface's width
- Set Object Height - Determine the string display surface's height

Conditions

- Text
 - Is Font Bold - Boolean return based on if Bold is set
 - Is Font Italic - Boolean return based on if Italic is set
 - Is Font Underline - Boolean return based on if Underline is set
 - Is Font Strikeout - Boolean return based on if Strikeout is set

Expressions

- Get Text - Returns currently displayed text
- Get Transparency - Returns value of transparency setting

Image

The Firefly Image object allows you to place on the highest level a 2d Image that displays over the Firefly engine rendered scene. Great for HUD elements. From the object's properties you can set multiple images to individual image frames and then call them at runtime.

Actions

- Images
 - Set Image to Display - Set which image frame you want to display via passing a value in the expression editor.
 - Load Image from file - Using an image selection dialog or a expression manually display a external image in the object
 - Load Image from Surface Object - Replace a image frame with a selected image from a Surface object
- Drawing
 - Set Destination X Coordinate - Set the upper left corner X position of image
 - Set Destination Y Coordinate - Set the upper left corner Y position of image
 - Set Destination width - Set object's the visible space horizontally
 - Set Destination height - Set object's the visible space vertically

Conditions

- Compare displayed image to a value - Compare the displayed image in a image object against a value using expression editor.

Expressions

- Images
 - Get displayed image index - returns the value of the currently displayed image frame
 - Get image width - Get the width of the currently displayed image frame
 - Get image height - Get the height of the currently displayed image frame
- Get object width - return objects set width of display space
- Get object height - return objects set height of display space
- Drawing
 - Get destination X coordinate - return the left corner X position
 - Get destination Y coordinate - return the left corner Y position
 - Get destination width - return the size of the visible space horizontally
 - Get destination height - return the size of the visible space vertically

Material Cache

The Material Cache object is used to store and array a collection of materials that are used to "skin" a variety of nodes. Once one has set the material image and settings, you can reference said materials numerical position in the "array" in a "set material" action is most node objects.

Actions

- Add a new material - creates a new material at the next available material index slot. Requires you give it a name using the expression editor.
- Copy a material - Allows using a materials index value to copy it's contents into another material slot by selecting the new slots index.
- Remove a material - Remove a material from the Cache by selecting the materials index value
- Edit Material - Allows you to edit the properties of a specific material
 - Set a material's name - allows you to set a existing material's name
 - Set a material type - Allows a user to set or change the display type of a material, kinda like a ink effect in some regards. For instance you can change it from Solid to Transparent Add. First select the material index value, then using a exact string match, identify which type to change to (Ex: "Solid")
 - Set a material's data type one - Sets an alterable value for a specific material
 - Set a material's data type two - Sets the second alterable value for a specific material
 - Textures (1 - 4)
 - Render Target - Whether or not this texture is a target for "Render to texture" action inside the Firefly Engine.
 - Render Target Texture Size -The size of the render texture.
 - Load Image from file - Using an image selection dialog or a expression manually display a external image in the material slot
 - Load Image from Surface Object - Replace a material with a selected image from a Surface object
 - Set Anisotropic Filter -

- Linear Filter - Sets the type of resampling / quality of the texture when resized.
 - Texture Wrap -
 - Set Mipmap Level of Detail Bias -
- Set Ambient Colour - The color when the object is in shadow
- Set Diffuse Colour - The base color of the material
- Set Specular Colour - The color of the object at direct light
- Set Emissive Colour - The color of the object that the material emits by itself
- Set Shininess -
- Set Thickness -
- Anti Aliasing - The type of anti-aliasing that will be used for the material
- Vertex Colour Material -
- Render Red Mask - Sets whether or not to render the colour red
- Render Green Mask - Sets whether or not to render the colour green
- Render Blue Mask - Sets whether or not to render the colour blue
- Frontface Culling - When on, the front of each face (outside of mesh) will be removed
- Backface Culling - When on, the back of each face (inside of mesh) will be removed
- Fog - When on, the material will be affected by the fog
- Gouraud Shading - When on, the material will smooth the light on each connected vertices
- Lighting - When on, adds light and shadow to the material
- Normalize Normals - Normalises faces after scaling
- Point Cloud -
- Wireframe - When on, lines between vertices are only rendered
- Z Write -
- Z Buffer - Sets how should the material be rendered, either in front or sent to back. Defaults to Less Equal.
- Use a Shader -
- Shader Settings -

Conditions

- On set shaders constants -
- First Texture is Render Target - Checks if the texture is a target for the "Render to target" action from the Engine object
- Second Texture is Render Target -
- Third Texture is Render Target -
- Fourth Texture is Render Target -
- Render Red Mask - Checks if "Render Red Mask" is checked
- Render Green Mask -
- Render Blue Mask -
- Render Alpha Mask -
- Is Frontface Culling Enabled - Checks if the front of each face is removed
- Is Backface Culling Enabled - Checks if the back of each face is removed
- Is Fog Enabled - Checks whether the material is affected by fog
- Is Gouraud Shading Enabled - Checks if the material smooths out faces
- Is lighting Enabled - Checks if the material has light and shadow
- Is Normalized Normals Enabled - Checks if the material normalises each face after scale
- Draw as Point Cloud -
- Draw as Wireframe - Checks if the material draws only lines between vertices.
- Is Z Write Enabled -

- Uses a Shader -

Expressions

- Get a Material - Returns the fixed value of a specific material, identified by index.
- Get number of Materials - Returns the number of materials currently inside the cache.
- Get a Material's index - Returns the index of the Material, found by its name.
- Get a Material's Name - Returns the name of the Material, found by its index.
- Get a Material's Type -
- Get a Material's Type Data One -
- Get a Material's Type Data Two -
- Textures -
- Get Ambient Colour -
- Get Diffuse Colour -
- Get Specular Colour -
- Get Emissive Colour -
- Get Shininess -
- Get Thickness -
- Get Anti Aliasing -
- Get Vertex Colour Material -
- Get Z Buffer -
- Get Loaded Vertex Shader string -
- Get Loaded Pixel Shader string -
- Get Vertex Shader Entry Point -
- Get Vertex Shader Compile Target -
- Get Pixel Shader Entry Point -
- Get Pixel Shader Compiled Target -
- Get On Shader Constant Material Index -
- Get On Shader Constant Node's Fixed Value -

Animated Mesh

With the Animated Mesh node you can insert an animated mesh into the scene. An Animated Mesh is animated across a number of Frames by calculating the position, scale, and rotation of the mesh's Joints using each one's collection of Keyframes across those Frames. So, an Animated Mesh might have many Joints, and each Joint can have many Key Frames of a Position and/or Scale and/or Rotation type. When the animation takes place, it calculates what Position, Scale, and Rotation each Joint should be set to at a given frame by working out each difference between the last and next Key Frame.

Current supported rigging modes are :

- .B3D files (skeleton mode)
- .MD2 and .MD3 (morph mode)
- .MS3D (skeleton mode)
- .X files (binary & text) (skeleton mode)

Firefly Movement

The Firefly Movement object lets you do complicated movements within the 3D scene easily.

You can use this object to assign Node objects Path and/or Directional Movements.

- Waypoints -
- Move a Node to - Select a Node via Fixed value and move it to a position using X, Y & Z inputs
- Move a Node with waypoints - Same as move a node to , accept you can add additional waypoints in front of the final end point. Using the following format in the expression editor, X,Y,Z|X,Y,Z|etc
- Move a Node in a Direction - Moving a node by setting Pitch, Yaw and Roll and speed
- Move a Node in the Direction of a Point - Move a node in a direction determining the Pitch, Yaw and Roll based on a position (X,Y & Z)
- Set a Node Movements Speed - Takes a node via Fixed value that is moving and allows you to change the speed.
- Pause a Node's Movement - Pausing a active movement on a node.
- Resume a Node's Movement - Return a node to a moving state after being paused.
- Stop a Node's Movement - Stops a node moving and cancels a previously assigned movement
- FPS Camera
 - Move a FPS Camera -
 - Move a FPS Camera With Gravity -
 - Set Mouse X Position - Moves the mouse cursor to given X screen coordinate
 - Set Mouse Y Position - Moves the mouse cursor to given Y screen coordinate
- On Collision > Handle a Node's Collision - This action will set a node's position and then will cause a check of collisions that may have happened. This action should really be used to set (handle) a node's position after a collision has occurred and use the node's coordinates of where it collided as parameters. This action will also handle a Camera's collision (should the node be a Camera), and set its Target to what it should be.

Conditions

- A Node has Movement - Comparing via Fixed value will trigger true if said node has a movement assigned to it.
- A Node's Movement is Paused - Returns true if a specific node's movement has been set to paused.
- Path Movement > A Node's movement is complete - Returns true if a node set to a path movement has reached it's final waypoint
- A FPS Camera is falling due to gravity - If the FPS camera is set to gravity and is in a falling state this will trigger

Expressions

- Get Speed of a Node's Movement - Returns the speed of a node that is moving
- Path Movement > Get Progress of a Node's Movement - Returns the progress of a node set to path movement.
- Calculations
 - Get Distance between two points - By feeding the expression two sets of X,Y, Z values you can return the distance value between them
 - Get 3D X Angle between two Points -
 - Get 3D Y Angle between two Points -
 - Get 3D Z Angle between two Points -
- FPS Camera Settings
 - Get X Position of FPS Camera - Returns X value from the position of the FPS Camera

- Get Y Position of FPS Camera - Returns Y value from the position of the FPS Camera
- Get Z Position of FPS Camera - Returns Z value from the position of the FPS Camera
- Get X Target of FPS Camera - Returns the X value of the point that the FPS Camera is looking at
- Get Y Target of FPS Camera - Returns the Y value of the point that the FPS Camera is looking at
- Get Z Target of FPS Camera - Returns the Z value of the point that the FPS Camera is looking at
- Get X Falling Velocity of FPS Camera -
- Get Y Falling Velocity of FPS Camera -
- Get Z Falling Velocity of FPS Camera -
- Get Mouse X Position - Returns the current mouse cursor position, in screen coordinate
- Get Mouse Y Position - Returns the current mouse cursor position, in screen coordinate
- Get Last Mouse X Movement - Returns the last frame's mouse position
- Get Last Mouse Y Movement - Returns the last frame's mouse position

Firefly Node - Particle System

With the Particle System Node you can insert a particle emitter into the scene. A Particle System node can be one of these types: Point, Box, Sphere, Animated Mesh Node, Cylinder, Mesh, Ring. These types describe the shape of the emitter.

What is actually emitted is similar to a Billboard node object.

A Particle System Node also can one or more Effectors. An Effector is something located at a X,Y,Z position within the scene and will affect particles generated by the node object in different ways. The different types of Effectors are 'Attraction', 'Fade-Out', 'Gravity', 'Rotation', 'Scale'.