

Machinima Maker

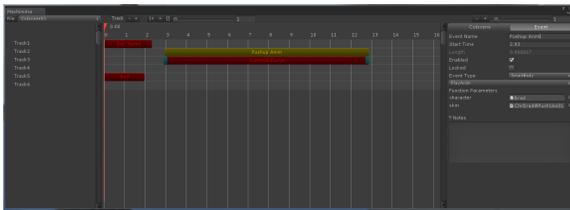
Page Contents

Overview

The Machinima Maker allows users to easily and rapidly create complex cut-scenes and vignettes from within the Unity Engine. Cut-scenes contain individual events on a timeline that are invoked when the cut-scene is played. These events can have any number of effects, from activating a game object in the Unity scene to sending BML and locomotion commands to [SmartBody](#). Many common events are provided out of the box, however custom events are allowed because almost any function written in a C# script can be invoked from the Machinima Maker.

Quick facts:

- Language: C#
- Distribution: Part of the `vhAssetsPackage` (`\bin\vhtoolkitUnity`)
- Platform(s): Windows



Users

Starting the Machinima Maker

The Machinima Maker is part of the [vhAssets](#) package. It is loaded from inside of the Unity Editor by clicking the "VH" menu on the menu toolbar and then clicking "Machinima Maker".

Creating a Cut-scene

If you have no previous cut-scenes in the Unity scene, then when you load the Machinima Maker window, there will be one single button that says "Add Cutscene."



If you have previous cut-scenes in the scene, you can add a new cut-scene by selecting File->New Cutscene

Adding/Removing Tracks

Click the - or + sign next to the "Track" text at the top center of the window. You can also right click in the area of the window where the track names are listed. Double clicking on track names allows you to rename them. Right clicking on track names allows you to add child groups.



Creating Events

There are 3 ways to create new events for a cutscene:

- Right click on the track area and select "Add Event"
- Ctrl + Left click on the track area

See below for an overview of the different types of events.

Selecting Events

- [Overview](#)
- [Users](#)
 - [Starting the Machinima Maker](#)
 - [Creating a Cut-scene](#)
 - [Adding/Removing Tracks](#)
 - [Creating Events](#)
 - [Selecting Events](#)
 - [Moving Events](#)
 - [Modifying Events](#)
 - [Setting Up Custom Events](#)
 - [Playing/Pausing/Stopping a Cut-scene](#)
 - [Automatically Playing Cut-scenes](#)
 - [Modifying Cut-scene Data](#)
 - [Creating Curves](#)
 - [Cutscene Prefabs](#)
 - [Event Types](#)
 - [Animation](#)
 - [Audio](#)
 - [Camera](#)
 - [Common](#)
 - [Renderer](#)
 - [SmartBody](#)
 - [Timed](#)
 - [Organizing Tracks](#)
 - [Using Hot Keys](#)
 - [Timeline](#)
 - [Tracks](#)
- [Developers](#)
 - [Creating New Event Types](#)
 - [Creating Custom Functions](#)
 - [Controlling a Cut-scene from Script](#)
 - [Controlling a Cut-scene from Unity](#)
 - [Listening for Cut-scene Callbacks](#)
- [Known Issues](#)
- [FAQ](#)

Note: This change isn't live in the latest Toolkit, however this works internally.

In order to select an event, left click on it. If you want to select multiple events at once, you can Ctrl+ left click them. To un-select an event, Ctrl + left click on it again. If you click and drag in an empty space, you can create and selection box which will select every event that it intersects with.

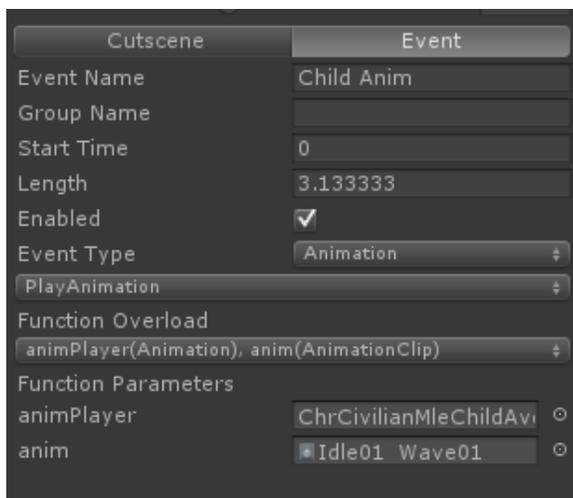
Moving Events

First select an event that you want to move by left clicking on it. You can now drag that event up and down to different tracks and left and right to change it's starting time. The keyboard arrows can also be used to move an event.

Modifying Events

Upon selecting an event with left click, you have access to the "Event" tab on the user interface. Here you can set event properties such as event name, start time, and event type. The event type is a category (such as Audio or SmartBody) that provides access to functionality that is inherent to the type (like playing a sound or using locomotion, respectively). Event types provide easy access to this functionality. See below for an overview of the different types of events.

Once the event type is selected, choose the functionality that you want the event to perform when invoked. Once chosen, make sure you fill out the required parameters listed below the selected type.



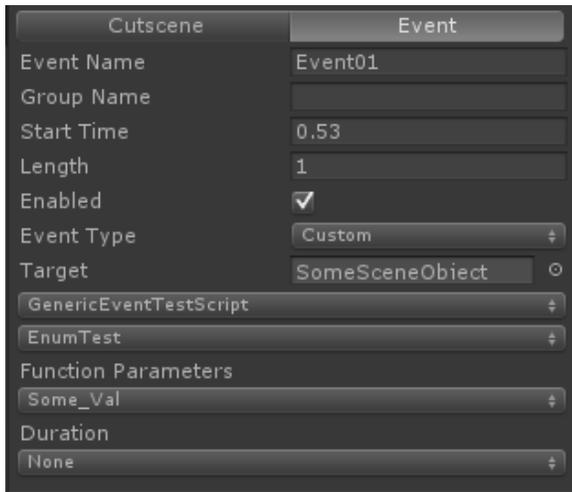
The screenshot shows the "Event" configuration panel. It has two tabs: "Cutscene" and "Event". The "Event" tab is active. The panel contains the following fields and options:

- Event Name: Child Anim
- Group Name: (empty)
- Start Time: 0
- Length: 3.133333
- Enabled:
- Event Type: Animation (dropdown menu)
- PlayAnimation: (dropdown menu)
- Function Overload: animPlayer(Animation), anim(AnimationClip) (dropdown menu)
- Function Parameters:
 - animPlayer: ChrCivilianMleChildAvi (text field)
 - anim: Idle01 Wave01 (text field)

Setting Up Custom Events

With custom events, your project-specific functions can be written and invoked through an event in the Machinima Maker. To set up a custom event:

1. Select an event and change the event type to custom
2. Drag the game object in the scene into the "Target" field
3. On the drop down menu, select the component that has the function that you want to invoke
4. On the next drop down menu, select the function that you want to invoke
5. Fill out the required function parameters



Playing/Pausing/Stopping a Cut-scene

Press the play button to start a cut-scene 

While a cutscene is playing, you can pause it by pressing the same button 

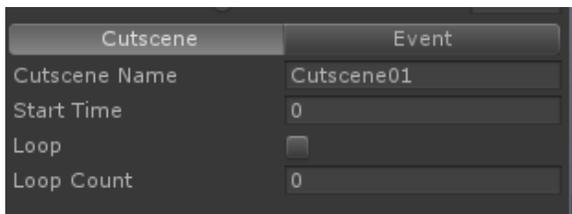
To stop a cutscene completely, reset all data, and reset the timer press 

Automatically Playing Cut-scenes

Attach the CutsceneManager script to any game object in your scene and make sure "Auto Play Cutscenes" is checked. When the Unity scene starts, all cut-scenes will play based on the start time that is listed under the cutscene tab. See [Modifying Cut-scene Data](#)

Modifying Cut-scene Data

You can change cut-scene data such as its name, number of times to play, and start time under the "Cutscene" tab



Creating Curves

You can create Bezier curves for Unity game objects to move along over a period of time using the Timed Event "Follow Curve." To create a curve:

1. Create a new game empty game object in Unity (Game Object -> Create Empty on the Unity menu toolbar.)
2. Add the "Curve" component to his game object. This object will now have a child game object named Point1, select Point1.
3. Hit ctrl + d on the keyboard to duplicate Point1. You'll notice when you do this, Point2 is created.
4. Move Point2 to a new location. Notice the yellow line being drawn between Point1 and Point2.
5. You can continue creating more points and moving them around.

Cutscene Prefabs

It is possible to create Unity prefabs for cutscenes. This would be the thing to do if you want to move cutscenes between scenes for example. If this is desired make sure to remember a few things:

- All event types should use *string* data for arguments. If they are linked to game objects in one scene those links would be broken in a different scene.

- Changes will need to be 'pushed' to the prefab by applying them, the same as what you would do with any other prefab.

Event Types

Animation

Animation events are related to Unity's animation system (not SmartBody's). If you want to play Smartbody driven animations, use the PlayAnim event under the SmartBody event Category.

Event Type	Purpose
Play Animation	Plays a Unity animation using the Unity Animator component.
Play Animation Queued	Queues the provided animation to play after the current animation is finished playing using the Unity Animator component.
Set Speed	Sets the play speed of the animation currently being played on the provided Animator component.
Stop Animation	Stops the currently playing animation on the provided Animator component.

Audio

Audio events are related to Unity's Audio Source component. If you want SmartBody character lip sync with audio, use the PlayAudio event under the SmartBody Category.

Event Type	Purpose
Play Sound	Uses the Unity Audio Source component to play the specified audio clip.
Set Pitch	Sets the pitch level of the specified Audio Source.
Set Priority	Sets the priority level of the specified Audio Source.
Set Volume	Sets the volume level of the specified Audio Source.
Stop Sound	Stops playing the current audio clip on the specified audio source.

Camera

Camera events are related to the Unity Camera component.

Event Type	Purpose
Set Camera Depth	Sets the camera draw ordering when multiple cameras are used in the scene. The lower the number, the higher the priority.
Set Field of View	Changes the field of view for the specified perspective camera.
Switch Cameras	Switches the current camera view to the specified camera.

Common

Common events provide easy to use functionality that is generally performed in Unity, such as moving, activating, and creating Game Objects.

Event Type	Purpose
Activate Game Object	Activates or deactivates the specified game object.
Apply Force	Applies a physics force to the specified rigid body.
Broadcast Message	Uses Unity's "BroadcastMessage" function to other game objects.
Create Game Object	Creates the specified game object in the scene.
Destroy Game Object	Deletes the specified game object from the scene.

Load Level	Loads a Unity scene.
Look At	Orients a transform to look at the specified transform.
Pause	Sets Unity's time scale to 0, freezing everything.
Play Cutscene	Plays a specified cutscene.
Rotate Game Object	Snaps a game object to the specified rotation. If you want smooth interpolation, see Timed Events.
Send Unity Message	Uses Unity's "SendMessage" function to all components attached to the specified game object.
SendVHMsg	Uses the VHMsg system to broadcast a message to other running processes
Set Parent	Sets the specified game object's parent in the scene hierarchy
Set Time Scale	Sets Unity's time scale to speed up or slow down updates
Transform Game Object	Snaps a game object to the specified position, rotation. If you want smooth interpolation, see Timed Events.
Translate Game Object	Snaps a game object to the specified position. If you want smooth interpolation, see Timed Events.

Renderer

Renderer events are for controlling Unity's Renderer component and it's associated materials and textures.

Event Type	Purpose
Enable Renderer	Hides or shows the specified renderer
Set Material	Sets the material to be displayed on the specified renderer
Set Material Color	Sets the color of the specified material on the specified renderer
Set Material Float	Sets the value of a float variable on the specified material
Set Material Texture	Set the texture to be displayed on the specified material
Set Material UVs	Sets the starting and size texture coordinates of the texture being used on the specified material

SmartBody

SmartBody events are for controlling SmartBody and SmartBody driven-characters.

Event Type	Purpose
Express	Sends out a vrExpress message. This can be used to play audio with lip sync
Gaze	Specifies a SmartBody driven character to look at a pawn or another character.
Gaze Advanced	Specifies a SmartBody driven character to look at a pawn or another character with many more parameters than the basic Gaze event
Move Character	Forces a character to start SmartBody driven locomotion.
Nod	Starts a head nod animation on the specified character.
Play Anim	Starts a SmartBody-driven animation on the specified Smartbody character. Note: the animation to play must be an .skm file.
Play Audio	Plays audio using an audio source and also uses lip sync.
Play FAC	Plays a facial viseme on the specified character
Play XML	Forces SmartBody to read the specified bml file and interpret it's associated .bml file and

	audio file in order to play audio and perform lip sync.
Posture	Sets the specified SmartBody character into the specified posture. Note: the posture must be an .skm file.
Python Command	Issues any arbitrary python command to SmartBody.
Rotate	Sets the yaw, pitch, and roll of the specified character.
Run Python Script	Forces SmartBody to open the specified python file and run it.
Saccade	Plays a saccade on the specified SmartBody driven character
Shake	Starts a head shake animation on the specified character.
State Change	Changes the current state of the specified SmartBody character to the specified state
Transform	Moves the specified character to the specified world x, y, z location.
Walk Immediate	Forces the character to start locomotion
Walk To	Forces the specified character to start walking towards the another character or pawn.

Timed

Timed events are interpolated over a period of time rather than being "Fire and Forget." The amount of time over which they occur is variables

Event Type	Purpose
Fade Audio	Fades the audio louder or softer on the specified audio source
Fade GUI Texture	Fades the current color and alpha of the specified GUI texture to the specified color
Fade Light	Fades the current color of the specified light to the specified color
Fade Renderer	Fades the current color of the specified renderer's material to the specified color
Follow Curve	Has the specified game object move along the specified curve
Rotate	Rotates the specified game object to specified orientation
Scale	Scales the specified game object to specified size
Translate	Moves the specified game object to specified location

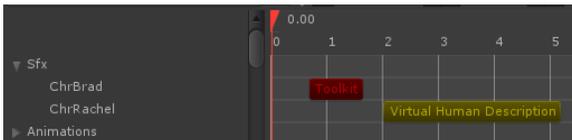
Organizing Tracks

Note: This change isn't live in the latest Toolkit, however this works internally.

Cut-scene tracks can be renamed and re-positioned in order to better organize the events of a cut-scene.

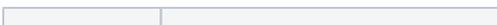
In order to rename a track, left click on the track name to select it, then left click again and you will be able to rename it.

Re-positioning a track or creating track hierarchies works the same as Unity's hierarchy view. Simply left click and drag a selected track and upon releasing the mouse button in an appropriate location, it will move.



Using Hot Keys

Timeline



Key	Function
Ctrl + Left Click	Create new event
Delete	Delete selected events
Up Arrow	Move selected events up 1 track
Down Arrow	Move selected events down 1 track
Right Arrow	Increase selected events start time
Left Arrow	Decrease selected events start time
Shift	Disables event vertical movement when dragging

Tracks

Key	Function
Up Arrow	Move selection up one track
Down Arrow	Move selection down one track
Right Arrow	Show selected track children (if it has any)
Left Arrow	Hide selected track children (if it has any)
Delete	Delete selected track

Developers

Creating New Event Types

In order to create your own event type category in the drop down selector, do the following:

1. Create a new C# class and derive it from GenericEvents
2. Override the GetEventType() method to return the name category of events that this class will hold. This is the name that will appear in the drop down selector on the gui.
3. Inside your class, create nested classes that derive from ICutsceneEventInterface. These classes will contain the functionality that is run when the event is invoked.
4. Inside each individual nested class, create 1 or more function overloads. You can call these functions whatever you like and they will be displayed on the gui when this event category is chosen.
5. Attach this C# script to the "GenericEvents" game object which is nested below the cutscene game object that you were working on
6. You should now see your new event type and the functionality that can be invoked from it on the gui. If you don't close the Machinima Maker and re-open it.

```
public class MyProjectSpecificGenericEvents : GenericEvents {
    public override string GetEventType() { return "This is the name
that displays in the event type drop down gui"; }

    public class MyProjectSpecificGenericEvent_SomethingAwesome :
CommonEvent_Base {
        public void DoSomethingAwesome(GameObject go) {
            // write your code
        }

        // this is a 2nd overload of the function (not necessary
to have multiple overloads, but this shows how)
        public void DoSomethingAwesome(GameObject go, float f) {}
    }
}
```

Creating Custom Functions

If you simply want to invoke a project specific function and you don't want to go through the steps involved in created new event types, then you can simple write a function in any C# script and select the "Custom" event type on the gui for the selected event.

Custom functions have a few rules:

- No return value is allowed
- No ref or out parameters
- No array, list, dictionary, or other collection types
- No structs

Follow these above rules and everything will work.

```
void CustomFunction(string output, int i, float f, Vector2 v2, Vector3 v3, Color col, bool b, Transform trans, GameObject obj) {}
```

Controlling a Cut-scene from Script

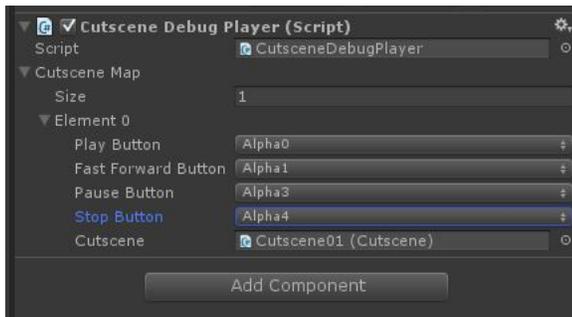
All cut-scene control functions such as pause, resume, play, and stop are exposed publicly for other objects to call.

```
public Class CutsceneController : MonoBehaviour {
    public Cutscene m_CutsceneToControl;

    void PlayCutscene() {
        m_CutsceneToControl.Play();
    }
}
```

Controlling a Cut-scene from Unity

1. Create an empty game object in the scene
2. Attach the CutsceneDebugPlayer script to it
3. Set the "Cutscene Map" size to how many cutscenes you want to control with keyboard keys
4. Set the cutscene that you want to control and the keyboard keys to control it



Listening for Cut-scene Callbacks

You can have project specific functions get called whenever a cut-scene is finished. To do this, define a function that follows this signature:

```
public delegate void OnCutsceneFinished(Cutscene cutscene);
```

Using a reference to a cut-scene, add your function to the callback list like this:

```
m_Cutscene.AddOnFinishedCutsceneCallback(yourFunction);
```

Known Issues

- Rewind doesn't always work.

- Stopping a cut-scene in the middle of a character playing a SmartBody animation will not cancel the animation.
- You can't rewind custom events

FAQ

See the [Main FAQ](#). Please use the [Google Groups](#) emailing list for unlisted questions.