

Introduction to Values in Multimedia Fusion 2

Introduction

Warning: You should already be comfortable with the frame editor and the event editor before starting on this tutorial, since I will not explain in details how you make objects and events in this tutorial.

Note: I will only refer to Multimedia Fusion 2 (MMF 2) in this tutorial. However, everything covered in the tutorial you can also do in The Games Factory 2. There should be no difference.

This tutorial will explain to beginners how we can use MMF2's features called alterable values, fixed values and flags. They may seem confusing at first, but once you learn how to use them, they will be invaluable in probably all your future games. Values are simple yet powerful tools both for adding dynamics into your game and to simplify your code. Values open up a lot of possibilities in your games. Without them there are many things you probably would like to put in your game that you simply cannot do.

Typical scenarios where you need to use values:

- Giving health to enemies
- Giving names and descriptions to objects in-game
- Turning on and of object states
- Storing positions
- Custom movements
- Visual effects

These are just some examples. Basically, values can be used in almost any situation.

There are 4 types of values in MMF 2. These are

- Fixed values
- Alterable values
- Alterable strings
- Flags

Fixed values.

What are they?

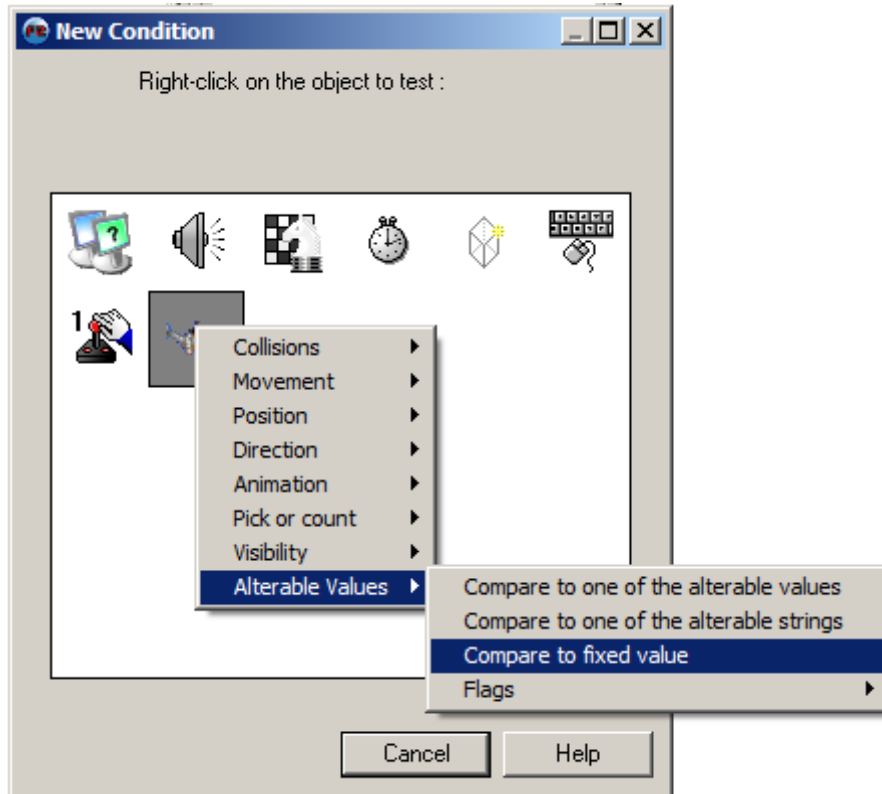
Fixed values are unique to every object in your game. You don't set this value yourself. All the objects in your game automatically gets a unique fixed value. Even if you have only one type of object in your game, all of the duplicates gets their own value.

How do I use them?

You can use fixed values to refer to a specific object. Here's a little example that shows how to first select an object, and then destroy it in another event.

- +User clicks on Apple

- - Counter1: Set current value to Fixed value of Apple
- +Upon clicking Spacebar
- +Apple: Fixed value = current value of Counter1
- - Apple: Destroy



Since you cannot set the fixed value of an object yourself, you will not find them among an object's actions. Instead, the fixed value are available to you when you make a new condition or use the expression editor to retrieve a value. In the above picture we make a new condition and want to compare the fixed value of the object.

Alterable values

What are they?

Alterable values are first and foremost, as their name is implying, values that are alterable. But they are more than that. Each object in your game, even duplicates, have their own alterable values. Each object have a total of 26 alterable values. They are called Alterable Value A, Alterable Value B etc... This means that you can actually store whatever value you want INSIDE each object.

A common use for alterable values are when you are giving health or making health bars for enemy objects. By using alterable values as health, each enemy may start out with a given set of healthpoints, and lose more and more health whenever it is hit, shot at, goes in a trap etc..

How do I use them?

Alterable values are really easy to use. You can set or get an object's alterable value at any given time. The object or objects that are present in the condition will be affected. You can also add to and subtract from an alterable value.

Here is an example of a health system:

- + Start of level
- - Spaceship: Set alterable Value A to 3

- + Bullet is overlapping Spaceship
- - Bullet: Destroy
- - Spaceship: Subtract 1 from Alterable Value A

- + Spaceship: Alterable Value A ≤ 0
- - Spaceship: Destroy

In the first event we don't pick a specific Spaceship in the condition, so all Spaceship will be affected by the action. All Spaceship now have an Alterable Value A of 3.

In the second event the action will effect the Spaceship that is overlapping the Bullet only. The bullet is destroyed and the Spaceship lose one health.

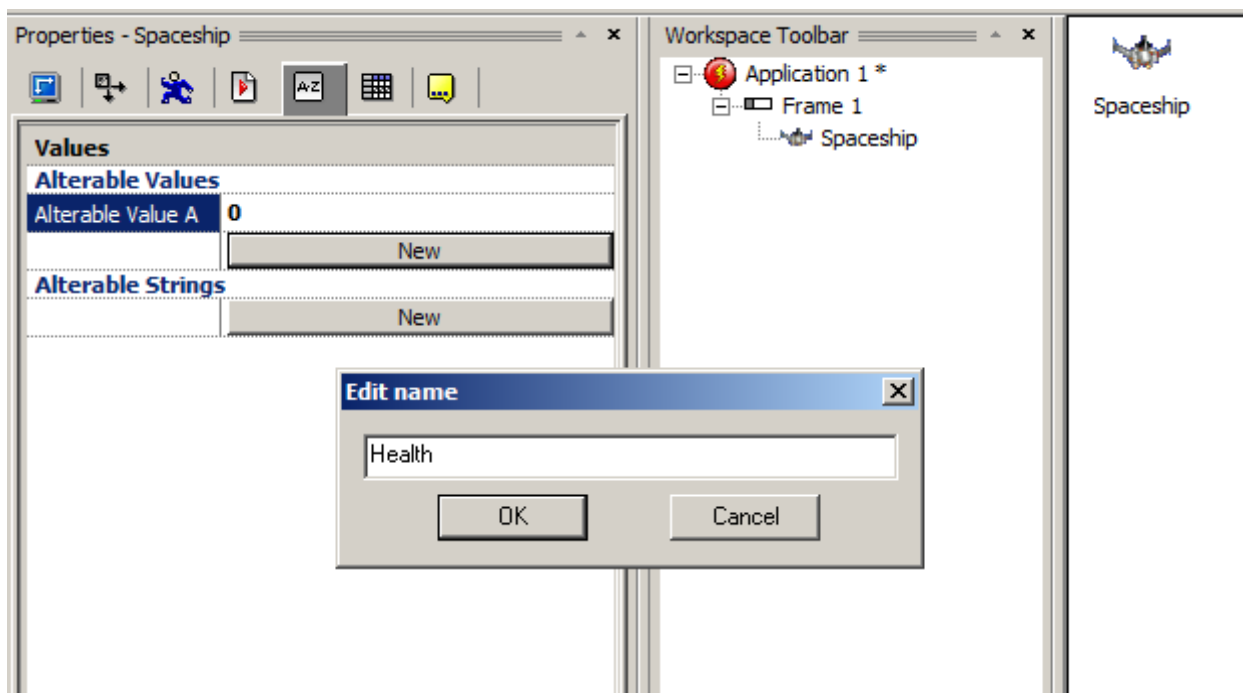
In the last event all objects with an Alterable Value A of 0 or less is affected. Any Spaceship that has gotten hit 3 times by a Bullet gets destroyed.

Naming alterable values.

In a big game where your objects have lots of alterable values, it will be hard to read your code if you don't rename the alterable values. You can do this in the Propertysettings of each object. Go to the frame editor, click on an object and select Properties. Then click on the Alterable Values Tab.

Click New to add a value. You can now select the alterable value and rename it.

For our little example above, it would be good to rename Alterable Value A to Health. You can rename the values at any time. All events that are using Alterable Value A will change the name to Health as well, so you don't need to worry about that.



Spreading values

There is one more action for the alterable values. Spread values allows you to number your objects. The value you spread defines the first number. Lets say you have 3 spaceships. If you spread value 0 into Spaceship, MMF2 will be numbering all Spaceships and start with 0. By default, the last object you created will get the value of 0, the next will get 1 and the next 2 etc... If you spread 5 instead of 0, the first Spaceship will get the value of 5, the next 6 etc..

This is especially useful when you need to loop through objects using fastloops.

Alterable Strings

What are they?

Alterable strings are almost the same as alterable values. The only difference is that they are text and not numeric values. They are maybe not as commonly used as alterable values, but they can be very useful if you want to store text that are unique for each object.

Imagine an adventuregame where you hover the mouse over different objects to see what they are called. If you have stored the name of the object in an alterable string, you can easily show their name in the game without having to manually hardcode it for each object.

How do I use them?

Alterable strings are used in the same way as alterable values, although for obvious reasons you cannot add, subtract or spread alterable strings. You simply set or get an alterable string.

Here is an example:

First make several active objects of different types and give them the Group.1 qualifier. Now edit each of the object's Alterable String A and set the text to the name of the Object. Then, enter the event editor.

- + Always
- - String1: Set alterable string to «»

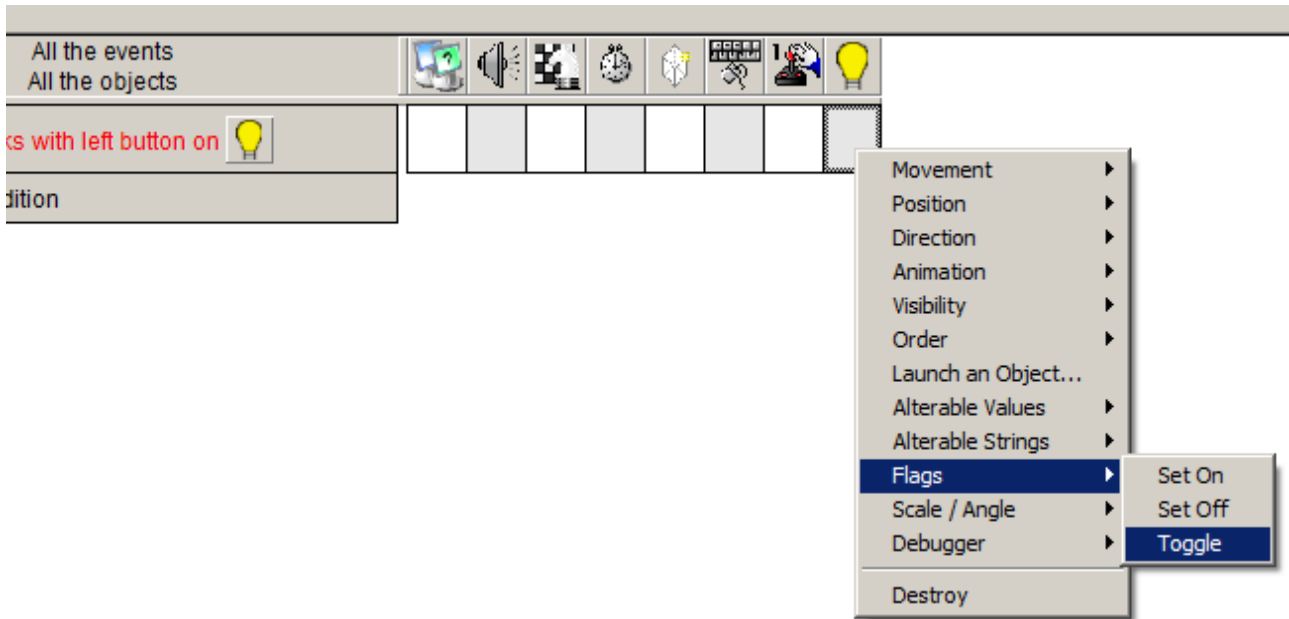
- + Mouse is over Group.1
- - String1: Alterable string = Alterable String A of Group.1

Flags

What are they?

If you think Alterable value were simple to understand, flags should be no match for you. They are basically the same as alterable values, only that they doesn't store a value, instead they are set either on or off.

A common thing is to imagine flags as switches. In programming languages they are often called bits or boolean values, that is either 0 or 1, False or True. In MMF2 they are either off or on.



How do I use them?

Flags are used in the same way as alterable values, only you have only two options: set them on or off. In addition flags have a great action that lets you toggle them either on or off, based on what state they currently have. Toggling a flag means that it changes its value to the other possible value. If it is off and you toggle it, it is set to on. If it is on when you toggle it, it is set to off. This makes flags better suited in some kind of situations than alterable values.

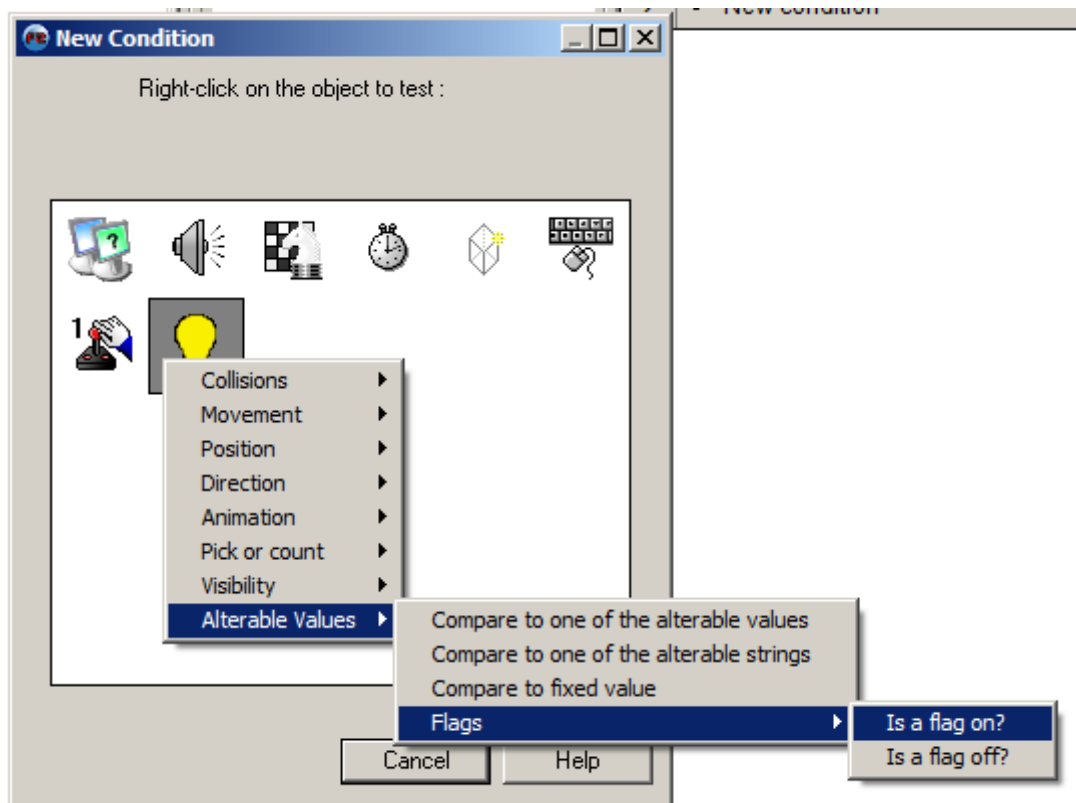
Example:

- + User clicks on Lightbulb
- - Lightbulb: Toggle Flag 0

- + Lightbulb: Flag 0 is off
- - Lightbulb: Set animation to animation «off»

- + Lightbulb: Flag 0 is on
- - Lightbulb: Set animation to animation «on»

Imagine how you could expand this example to make a room darker when the light is off, and brighter when the light is on.



Credits

This tutorial was written by Kjetil Nossun alias Popcorn from the Clickteam forums. For any questions or feedback, send a mail to ilikepopcorn@hotmail.com or send a private message to Popcorn on the clickteam forums.

2010 May 22