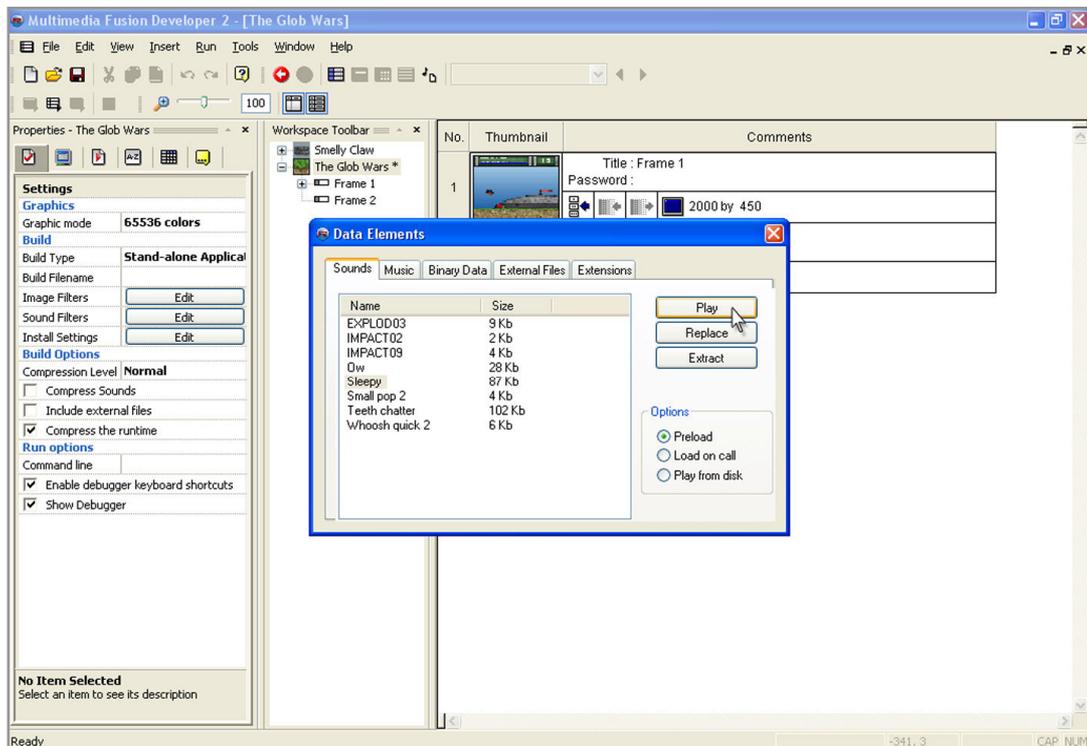


THE MULTIMEDIA FUSION 2 INTERFACE GUIDE



You may not use this tutorial for any other purpose than learning, working or having fun... In other words: You can use this tutorial for anything You'd like, as long as it doesn't involve both a hammer and a squirrel.

Hi there, all!

Welcome to the Multimedia Fusion 2 Interface Guide, teaching you how to control and efficiently work with the best multimedia authoring tool ever - [Multimedia Fusion 2](#) by Clickteam! This tiny tutorial will give you an overview of the interface and basic features of MMF2, and is highly recommended for users that are new to the fascinating world of Clickteam's products. The main purpose of this guide is to help you discover how easy it is to create your own games, applications and presentations with Multimedia Fusion 2 – whether you believe it or not, MMF2 enables you to create astonishing projects with just a few clicks and a bit of “drag & dropping”! And yes, it's true - no programming skills required!

If you are already familiar with Multimedia Fusion's functions and interface, skip this tutorial and go for the other ones that can be found on Clickteam's website (such as the *Smelly Claw* tutorial, the *Glob Wars* one or the *Enhancing the Feel* tutorial series). If you have any questions that need answering – don't hesitate to join Clickteam's forums and ask about things there. You'll soon discover that the Click community is always willing to help, and that no questions are left unanswered.

What you'll find in this tutorial - List of content:

- *MMF2 startup & updating*
- *An overview of the main interface*
- *The Storyboard Editor*
- *The Frame Editor*
- *The Event Editor*
- *Testing an application*
- *Koobare's MMF-to-Paper coding system*

If you have any problems with this tutorial, or notice that there are some mistakes present, please, contact me and I'll do my best to help you and replace all the errors with correct information.

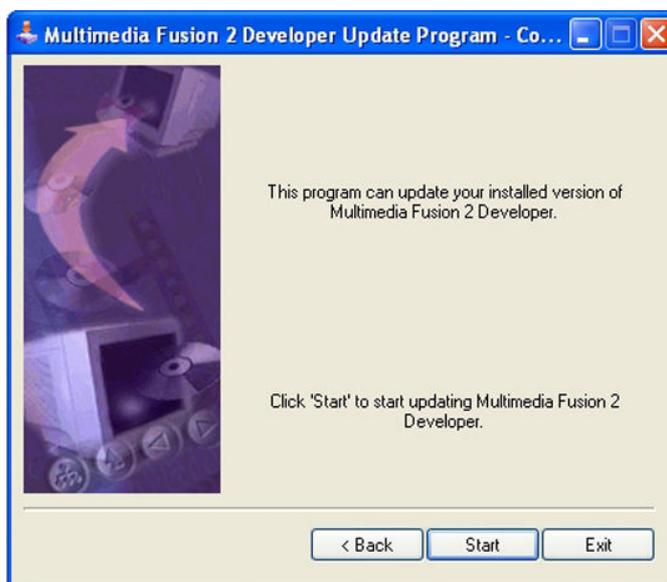
Contact me at: marchewkowy@gmail.com

Part I: MMF2 startup and updating

OK, so you're ready to start your adventure with MMF2, are you? Start Multimedia Fusion 2 by double-clicking on the program's icon or its shortcut. During the loading time, an introductory information window will appear, showing you such data as the program's version, build and registration info (see image to the right). During the loading sequence MMF2 will connect to Clickteam's server and check if a new build is available (this only happens if you're connected to the internet and haven't disabled the "check for updates" option in the preferences).



If a **new update is available**, MMF2 will ask you whether you'd like to install it or not. Since having everything up-to-date is always a good idea, I'd suggest clicking "yes" every time such window appears. If you do so, MMF2 will download its newest update (this sometimes takes a while) and open its setup file – once the update is installed and setup exits, Multimedia Fusion 2 will continue with loading.

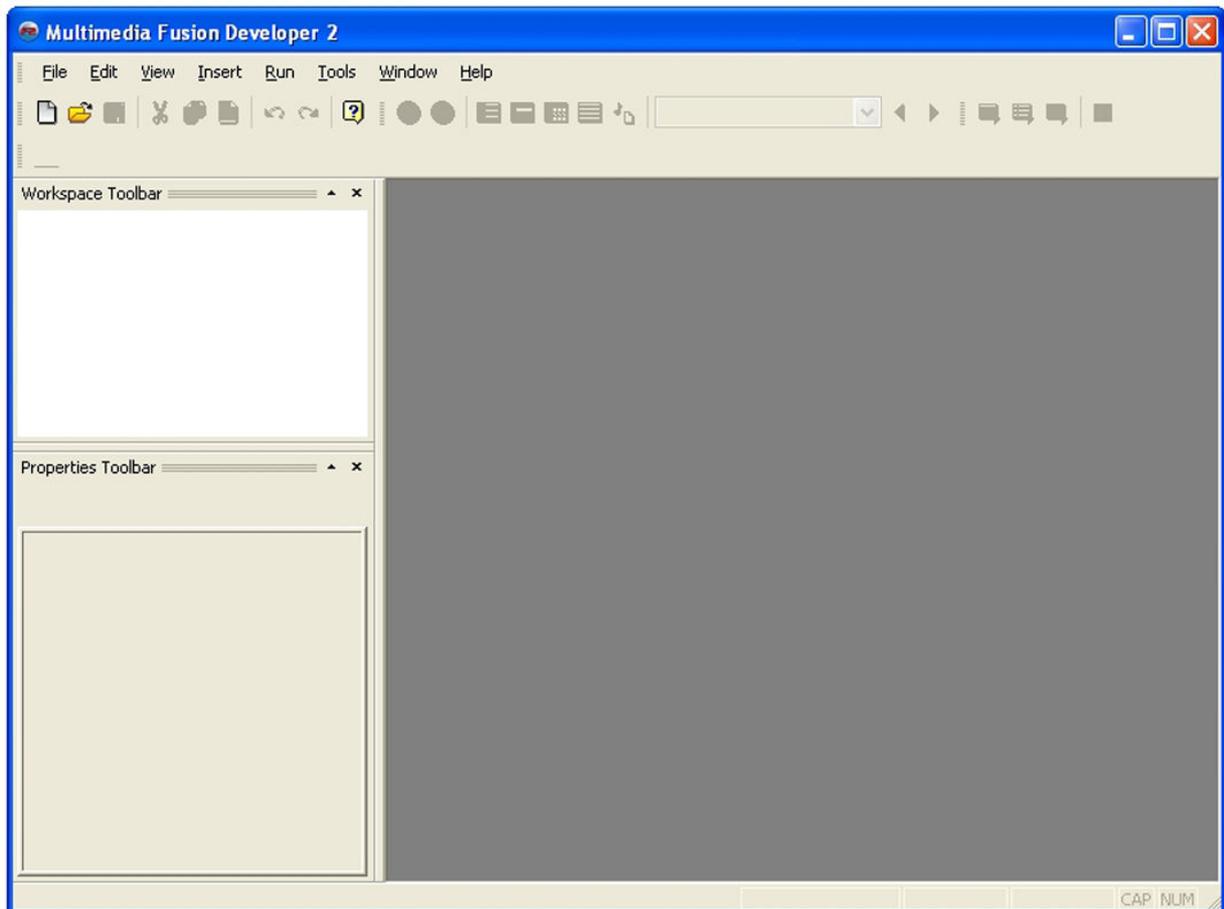


Additional info:

- If you don't want MMF2 to connect and check for updates on start, you can disable this function in the **Preferences** (in the "general" tab).

Part II: The main interface

Once we're through with the startup phase, Multimedia Fusion 2 will be initiated and you will see the main screen interface. It should look something like this:



Looks nice, doesn't it? Let's see how to use it in action. We need to create a new application. To do so, either select the "**New application**" icon (📄) from the Main Toolbar, select the **File > New** option from the menu or use the "**Ctrl+N**" keyboard shortcut (try to remember this one as keyboard shortcuts can be quite useful and speed up you up a bit).

method 1



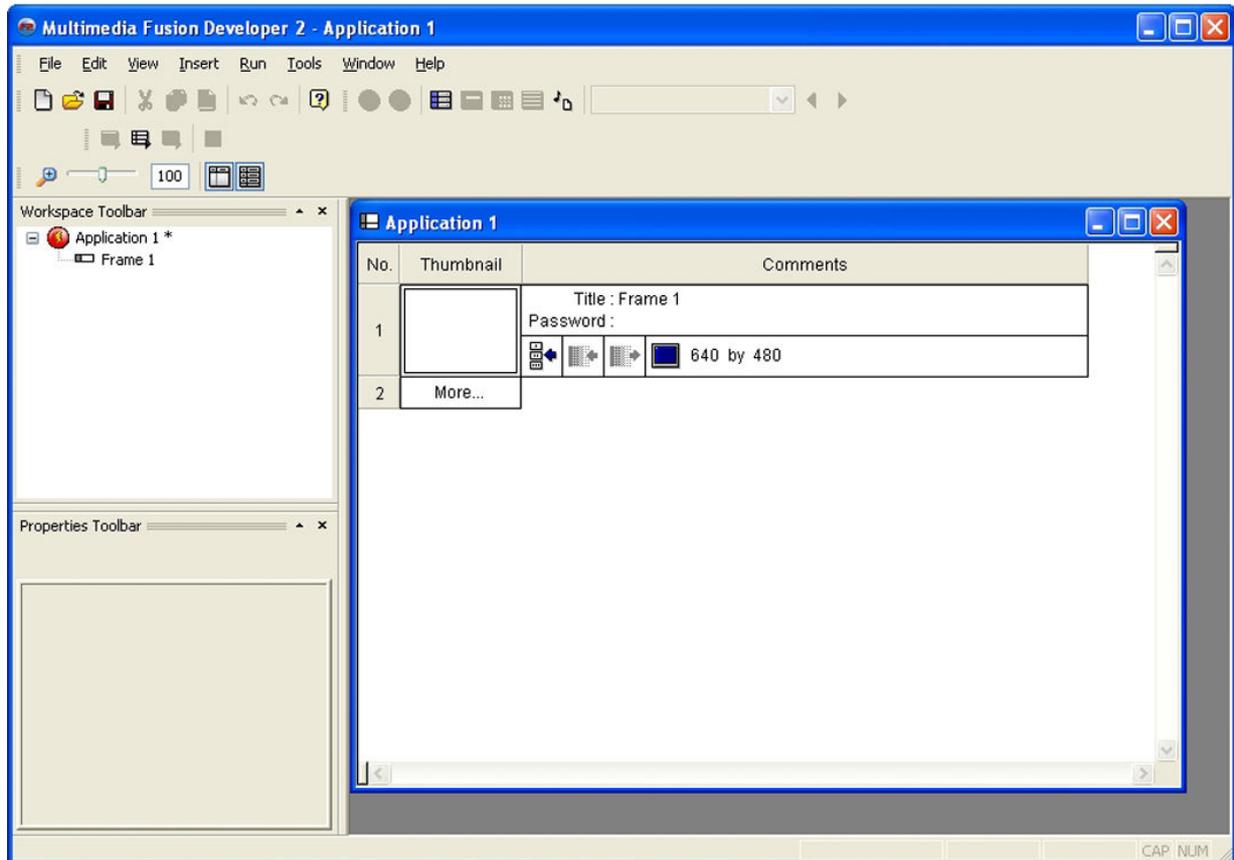
method 3



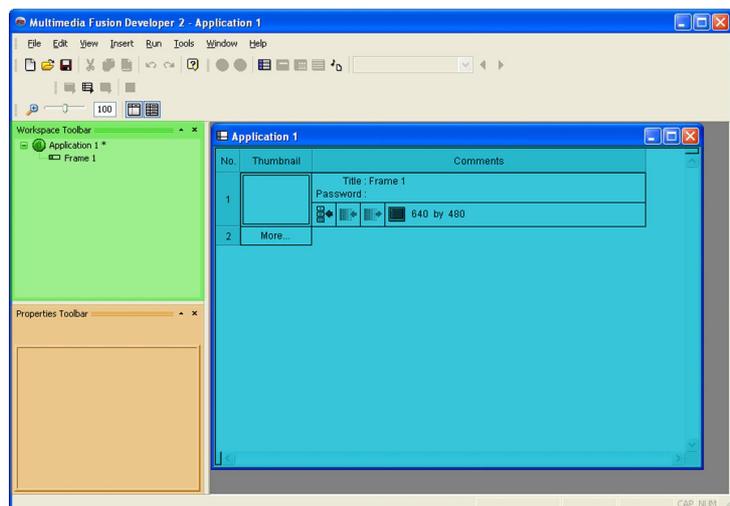
method 2



OK... We've just created a new application. Note that the main interface changed a bit: a couple of previously unavailable buttons from the menu toolbars became active, a new window named "Application 1" appeared in the middle of the screen... Here, take a look:



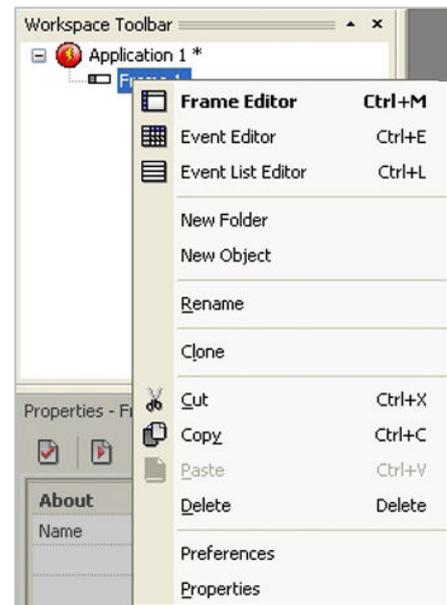
Notice, how the main screen is divided into separate areas: the upper menu with all the smaller toolbars, **the workspace toolbar**, **the properties toolbar** and the **main work area**, currently having the "Application 1" Storyboard Editor window on display. Just take a look at the image to the right – the workspace toolbar has been highlighted green, the properties toolbar is orange, whereas the Storyboard Editor window is blue. You can rearrange all these elements in any way desired, by simple dragging and dropping them around the screen.



Let's take a closer look at the **Workspace Toolbar** (you can see an example one on the image to the right). It's primary function is to display all the contents of your newly-created application. Here you can view and browse projects, applications, their frames and objects that are contained within the specific frames – all in a neat and clear tree-like view. Since a usual MMF2 application is actually a set of connected frames, tied together with global values, actions and other pieces of code, you can bet that you're going to use the workspace toolbar a lot!



Now, let's return to the still fresh application that we created a moment ago. Move your cursor to the workspace toolbar and right-click on "Frame 1". A drop-down menu will appear, containing a list of commands – take a look at the screenshot to the right. The first three commands from the list enable you to quickly change between editors that are to your disposal – the **Frame Editor** (where you create and set up all the objects that are going to be used in your application), the **Event Editor** (where you can code the events that make your application "alive") and the **Event List Editor** (which is an alternate way to code your events - display of the events in this editor is done in a list, with conditions and actions listed one by one).



The next two commands enable you to create new objects inside the selected frame (the "**New Object**" command) and to organize them into folders (the "**New Folder**" command) – I never actually found these two to be useful on the menu, since they can be accessed a lot easier inside the Frame Editor itself. Moving on downwards, you can find commands that enable you to rename, clone, cut, copy, paste and delete parts of your app (both frames and objects), and – last, but not least – two commands that enable you to quickly access the **Preferences** of the Workspace Toolbar and the **Properties** of the selected frame. Select the "Properties" command from the menu and take a look at the **Properties Toolbar** – it should now display the properties of our first frame (note that usually you don't have to select the menu command to show the object's or frame's properties – if the Properties Toolbar is visible, merely selecting the object will open its settings and enable you to mess a bit with them).

Additional info:

- As most of the MMF2 interface, the **Workspace Toolbar** enables you to easily drag & drop the elements it's displaying. Thanks to the drag & drop method you can easily move or change the sequence of objects or frames if you wish. You can even drag them between applications or project files!

Now, let's move on to the next thing on our "to-do" list... **The Properties Toolbar** (take a look at the screenshot to the right) is yet another interface element that you're going to use a lot, since it is used to show and edit the properties of the selected application, frame or object. Whenever you're going to add a fade-in animation, change the object's movement or make the frame's virtual width bigger – this is the place you're going to end up messing with.



Now, take a look at the Properties Toolbar that you have in front of you. It seems pretty well designed, doesn't it? The icons at the top of the Toolbar enable you to choose the categories of properties that you currently want to play with. The first one, called "Settings" enable you to change things such as size, background color, palette, fade-ins and fade-outs, ink effects, transparencies and a lot, lot more, differing on the object that we're currently editing. The second button opens the "Runtime options" tab, the third – with the yellow dialogue balloon – show you the "About" properties of an object. Differing on the object that we're examining, there can be also many more tabs to choose from – such as "Movement", "Values" or "Events" tabs. When you pick the settings category you're interested in, just click on the property you want to change and – either by clicking the "edit" button, typing in new values or selecting an option from a drop-down menu – just change it to whatever you'd like to.

Would you like a little repetition to memorize all this info? Hmmm, let's see what the good ol' MMF2 Help file has to say about this: "*The Properties Toolbar, as its name indicates, shows*

the properties of the currently selected object. This object can be an application, a frame or an object in the frame. Each property is shown as a single line, with means to edit it. The edit options can be a button, a checkbox, a color selector, an edit line etc.: every line can be different from the other depending on the item to edit. To edit an item, simply click on it. The properties are grouped in folders, and you can choose the correct folder by clicking on the tab icon on the top of the properties toolbar". Just try it out now, if you wish, by going to the "Settings" tab (the first one to the left) of your Properties Toolbar, changing the background color to black and then returning it to white again. Isn't it easy? Sure it is. Let's move on, then...Onto part three! MMF2 Storyboard Editor, here we come!

Additional info:

- You can easily **dock** and **undock** both the Workspace and Properties Toolbars, moving them around the screen for your convenience. To undock a selected toolbar, just drag it to the center of the screen.

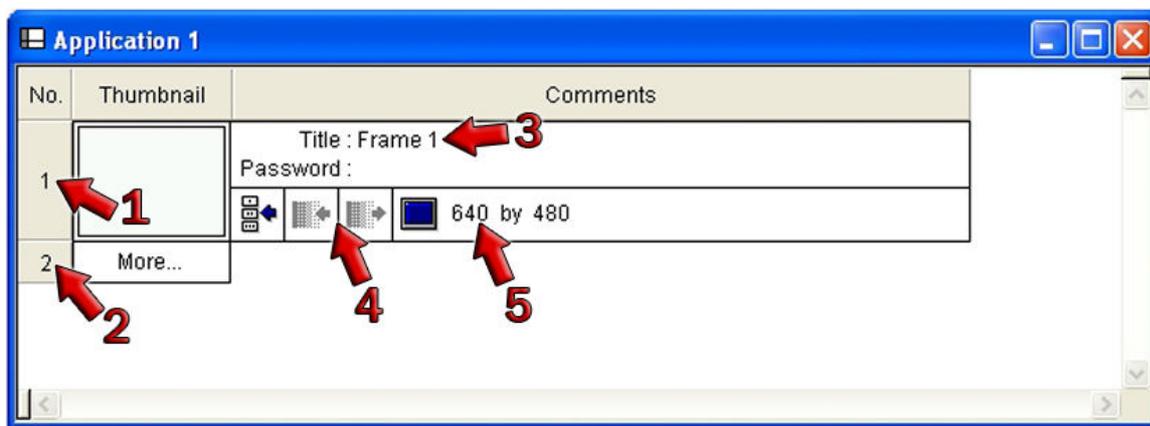
Part III: The Storyboard Editor

We earlier mentioned three MMF2 Editors – the Frame Editor, the Event Editor and the Event List Editor. Now we're going to examine the fourth one – the **Storyboard Editor** (take a look at the screenshot below for basic visual orientation).



The Storyboard Editor is where you edit the sequence and basic properties of the frames of your application. If you wish to create new frames in your app, change their order, resize them, change their title or add a password to one or two of them – this editor is the place to go. You can also access most of these settings from outside the Storyboard Editor (as they are available via the main menu or the Workspace Toolbar), but I prefer using the SE, since it keeps everything as simple as possible, in a clear “all-in-one-place” way.

Let’s take a closer look...The screenshot below shows us some points of interest in the Storyboard Editor’s interface – their description is presented under the image, just follow the red arrows and then read the corresponding explanation:



1) This is a **frame number**, each created frame has one of it’s own, depending on the sequence. Left-click on it to open that frame in the Frame Editor. Right-click on it to show a drop-down menu with rename, clone, cut, copy and delete functions.

2) This is a frame number that will be given to the next frame you’ll create. Click on it to create a new blank frame.

3) Click here to change the **title of the frame**. Right below you have the possibility to change the **password** – just click right by the “Password” text and type in the password you’d like to assign to this frame.

4) Here you can add and edit the **fade-in and fade-out frame transitions**. Click on of the buttons to open the transition setup, where you can add a transition and control it’s preferences.

5) Click on one of the edit boxes to change the **width** and **height** of your frame.

Between the frame number and the comments section you can find the **thumbnail image** – it shows you how this specific frame looks like in the Frame Editor, so it can be useful when you’re looking for one of the levels in a multi-frame application. Left-click on it to select the frame, double-click to open it.

One more thing is worth noticing when it comes to the Storyboard Editor – and that’s the **Storyboard Editor Toolbar** that’s docked into the upper part of the interface by default. Look for such a toolbar while you’re at the Storyboard Editor:



This small toolbar makes it a bit easier to control the whole Storyboard Editor – you can easily zoom the SE in and out (just use the slider situated right by the magnifying glass) and decide whether to show or hide the headers and comment section. This isn’t really a big deal, but it can help you a bit to better organize your work.

Well, that’s all when it comes to the Storyboard Editor... Let’s move on. Next stop: a thing or two about using the **Frame Editor!**

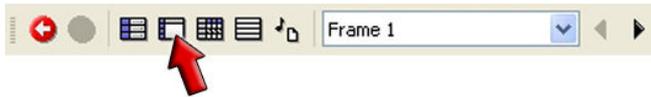
Additional hint:

- Remember that both the drag & drop technique and standard system keyboard shortcuts (*Ctrl+C*, *Ctrl+V*, *Ctrl+X*) work in the Storyboard Editor.

Part IV: The Frame Editor

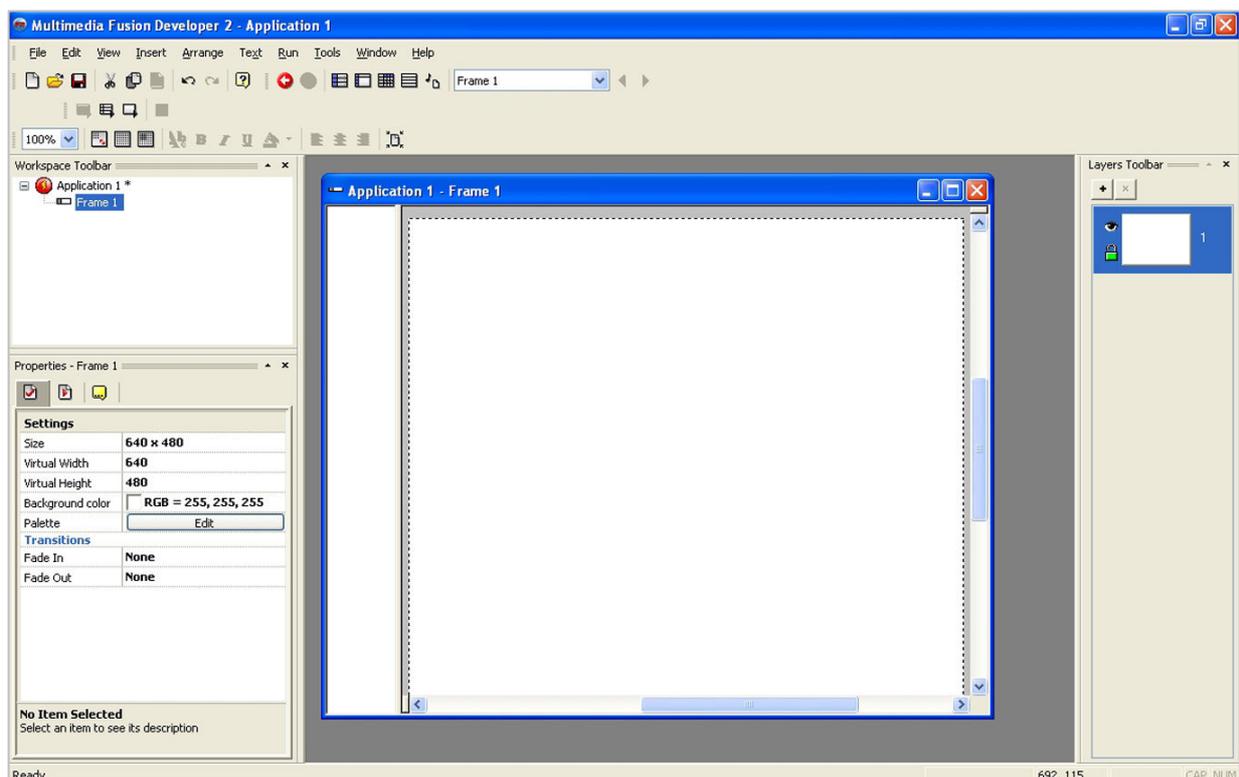
There are dozens of ways to access the Frame Editor – I guess that you’ll decide which one to use later on, once you’ll get a grip on all MMF2’s primary functions and controls. One of the easiest way to get to FE is by right-clicking on the frame we want to edit in the Workspace Toolbar, and then selecting the “Frame Editor” command from the drop-down menu. Not a satisfying option for you? Then select the desired frame (either in the Workspace Toolbar or in the Storyboard Editor) and press **Ctrl+M** on your keyboard. Still not good enough, huh?

Maybe you'll find this one interesting: go to the top menu, select **View > Frame Editor**. You can also left-click on your frame's number in the Storyboard Editor (or double-click on the frame's thumbnail). And – last, but not least – there's always the good ol' way of opening the Frame Editor by using the **Navigation Toolbar**. To do so, just find your way to the button pointed out at this image:



Wasn't to hard to find it, was it? The **Navigation Toolbar** can come quite handy too, you know. It not only gives you the possibility to quickly cycle between the MMF2 editors (The Storyboard Editor, the Frame Editor, the Event Editor, the Event List Editor, the Data Elements screen), but also enables you to travel between frames of your application, either by choosing the specific frame from the drop down menu, or by using the "previous frame" / "next frame" buttons (note: you can also cycle between frames by pressing **Ctrl+LEFT** and **Ctrl+RIGHT**).

Anyway, let's set all this theory aside – just open your **Frame Editor** and let's get this show on the road. Once you're in the Frame Editor, you should see something like this (note that your interface layout can differ a bit from the one presented here – don't panic if it does, the whole interface is pretty configurable and you can drag & drop most of it's elements):



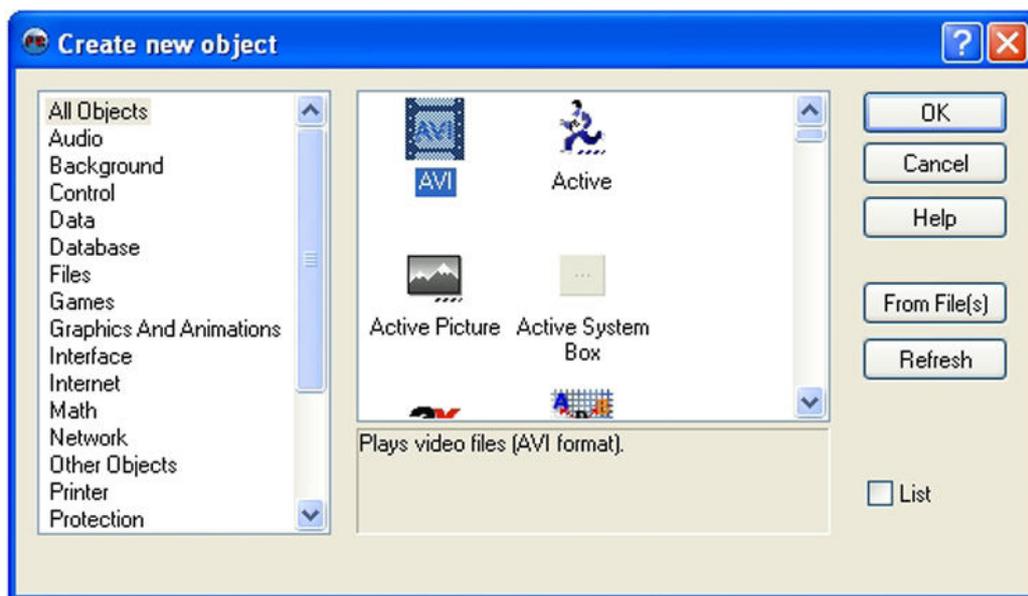
First of all, let's maximize the Frame Editor window, shall we? Press the  button in the right-top corner of the window. This isn't really necessary, I just prefer to work that way.

Second of all – let's introduce ourselves to a bit of theory. What is the Frame Editor exactly? The MMF2 Help file has an answer: *“The Frame editor is one of the most important editors in Multimedia Fusion 2. It allows you to dispose all the objects of a frame of your application nicely, and create your display progressively. The Frame editor consists of 2 panes. The left pane, usually small and vertical, is called the Object List window. It lists all the objects used in the frame. The right pane displays, and allows you to edit, the content of your frame. Your frame is displayed in the middle of the edit area, and is surrounded by a gray area. The gray area is not displayed at runtime, you can use it to hide objects. You can change the size of the gray area in the preferences of the Frame Editor (Horizontal and Vertical Margins)”*.

Wow, that's some load of information, isn't it? Let's see it in practice, then. Let's use the Frame Editor to create a new object – let's say, an **Active Object**.

To do so, either select the **“New Object”** command from the **“Insert”** menu, or right-click somewhere in the Frame Editor (you can also right-click on the specific frame in the Workspace Toolbar) and select the **“Insert Object”** command.

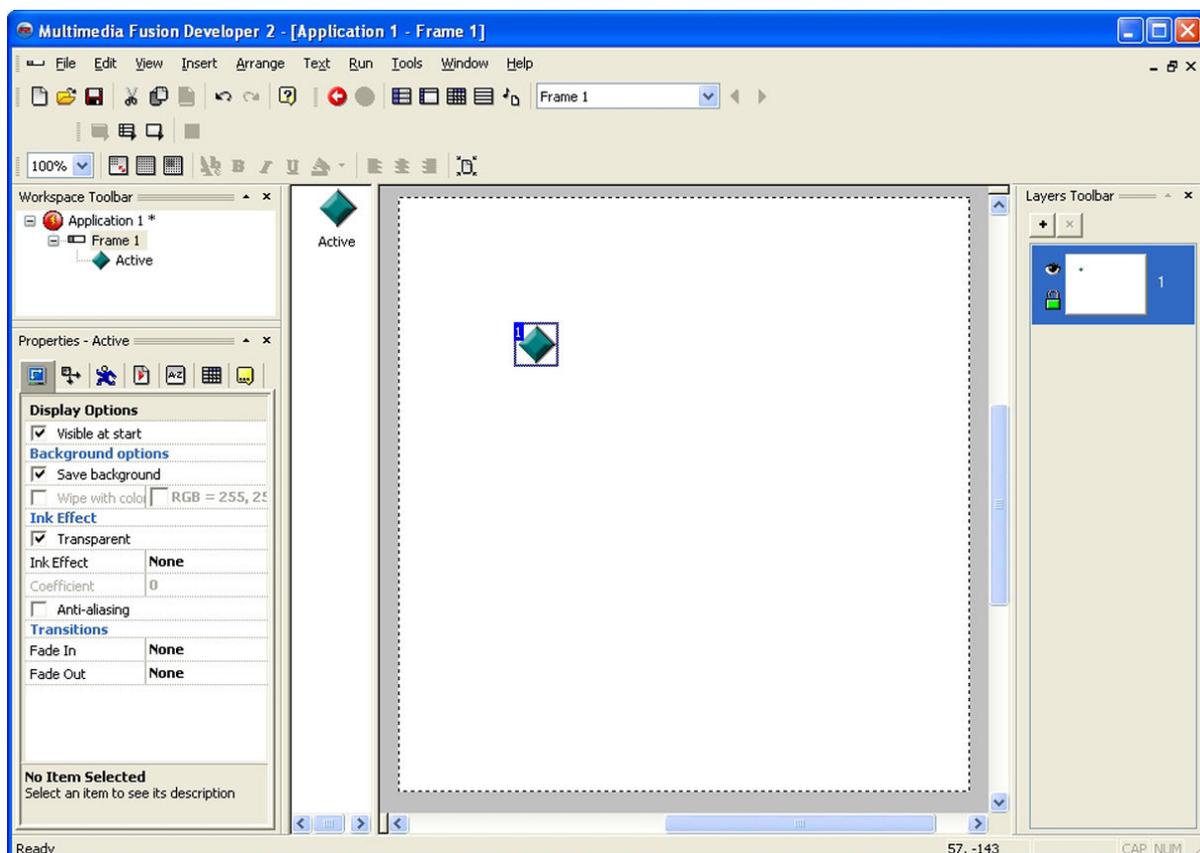
The **“Create new object”** window will appear:



All you have to do here is to select the **“Active”** object from the Objects List (that's the one with the icon of a little man running) and click the **“OK”** button. But – since we're not in a hurry –

don't do that yet. Firstly, let's examine this window a bit. The **“Create new object” dialog** is displayed every time you insert a new object into one of your frames. It's main purpose is to allow you to choose which object would you like to create. The list to the left is a list of object categories – you can choose here which type of objects should be displayed in the object browsing area (the right pane). You can also import objects from other Multimedia Fusion 2 projects and applications via this window – just click the **“From File(s)”** button and an explorer window will open, enabling you to select the files from which new objects will be imported. Just a few millimeters below another button can be found - the **“Refresh”** button. MMF2 enables you to add new extensions even while the program is running (sic!) – the only thing you have to do, is to push the “Refresh” button after the new extension has been added, just to include the new object in all the lists and categories. There is also a small checkbox named **“List”** at the bottom of this window – check it to display all the objects in a bit smaller form.

When you're done looking around, select the Active object and press **“OK”** (you can also double-click on the Active object - result will be the same). The “Create new object” window will close and your mouse pointer will change itself into a precision crosshair. Just click somewhere in your frame to place the newly created object there, and... Voila! We've just created our own little Active object! Your MMF2 should look a bit like this right now (don't worry if it doesn't look identical):



Notice that our object was added not only to the frame, but also to the previously mentioned **Objects List**. Notice also, that when you select the object from the list, it's counterpart in the frame is also being selected – that's why the **Objects List** can be a really, really helpful device, especially when you have a lot of objects laying around. When your project is really getting big, it's always easier to find a specific item on the list, than in the frame, where it could have been layered below another object – and let's not forget that you can arrange Objects List's icons by name or by type (just right-click somewhere on the list to open up a drop-down menu, go to the "Arrange Icons" submenu and select the option you'd like to use).

Now, let's get back to our little **Active object** for a moment. Deselect all objects by clicking somewhere in the frame, but not on the object itself (you could also select a command from the top menu: **Edit > Unselect**). Then, click on the Active object – notice how it becomes surrounded with a blue rectangle, also notice the number in the upper-left corner. Now, once more click (without unselecting!) on our little object – notice that 8 dots appear around it. They are used for quick resizing, and can really come in handy sometimes – let's try them. Just select one of them with a left-click, hold your mouse button and move your mouse a little to witness the resizing process.

That was easy, wasn't it? Press "**Ctrl+Z**" or select the "**Undo Resize**" command from the "Edit" menu to return our object to it's previous size. Now, click once more on the object – those little dots should be replaced by four bent arrows – this means that our object is in the rotation mode. Just grab one of the arrows with your mouse and move it around to rotate the object. Once your done, once again **Undo** all changes to the object's appearance.

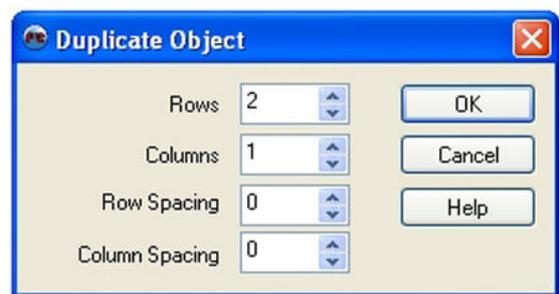


OK, so we now know how to rotate and resize our object. But there's a lot more things that we can do to it, some of which are quite complicated, others are not. Firstly, let's give our Active object a more unique name, shall we? Right click on it, so that a drop-down menu appears. Select the "**Rename**" command. Input "*RumbleBumble*" and click OK.

Take a look at our frame. Doesn't it look a bit empty and unused? Sure it does. We need more objects. And that's the perfect excuse for me to introduce **cloning** and **duplicating** to you – two methods of copying an object. What's the difference between them, you ask? **Cloning** creates a totally new object, with the same properties and looks as the original one – but it's a

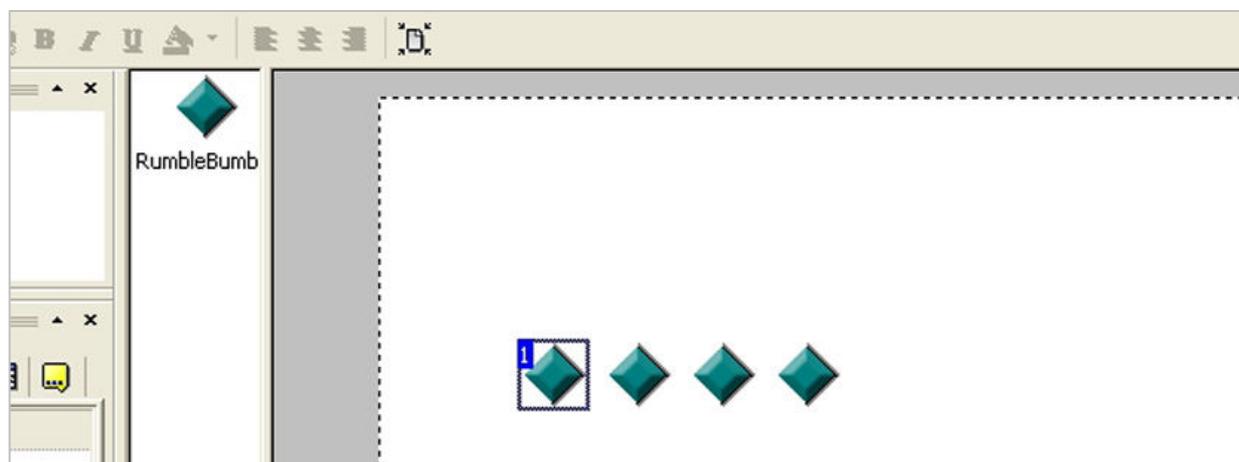
different object, has a different name, you have to create different events in the Event Editor for each cloned object. **Duplicating** is a bit different – it creates a new occurrence of the same object – if you’ll change the settings on one of the duplicates, resize it or add a new ink effect – **all this changes will be applied to all of the duplicates, as well as to the original object**. This is pretty important, so keep this in mind. Both these options are available in the object’s drop-down menu (just right-click on the object you want to copy and select either the “Clone object” or “Duplicate” command), or in the “Edit” menu. You can also quickly duplicate an object by using the **Ctrl+C** and **Ctrl+V** keys or by dragging & dropping objects from the Objects List (just select an icon in the list and drag it onto the workframe – a duplicate will be created exactly in the spot where you dragged the icon to).

All of this seems a bit complicated? Don’t worry, we’ll practice this in a second. Let’s say that we want to have **three duplicates, two clones and the original object** – that will give us six objects in total. Right-click on our *RumbleBumble* object. Select the “**Duplicate**” command. The “**Duplicate Object**” dialog



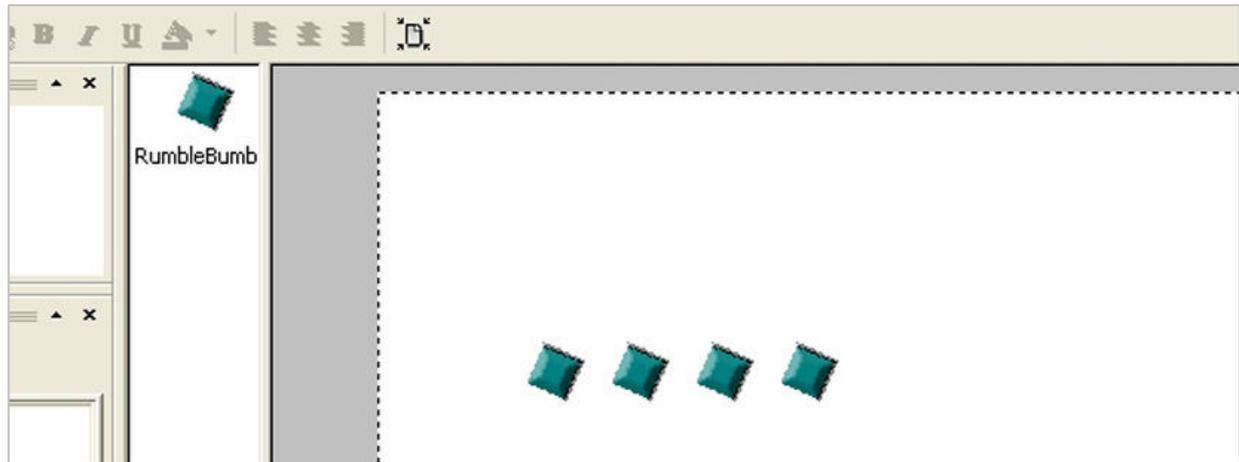
window will appear (take a look at the image to the upper-right). Now we have to enter the number of rows and columns of objects we want to create, and how many pixels between them horizontally (“Row Spacing”) and vertically (“Column Spacing”) do we want. Let’s set the columns number to 4 (that will give us the original + three duplicates), the rows number to 1, set the “Column Spacing” to 10, leave the “Row Spacing” at 0. Click OK.

This is what we’ve got now – a row of four duplicated objects:



Note that there’s still only a single “RumbleBumble” in the **Objects List**!

Time for a little experiment! Select one (and one only!) of the four objects available and change it's size or rotation. For example, I've rotated the first object in a row... And here's what I've got:

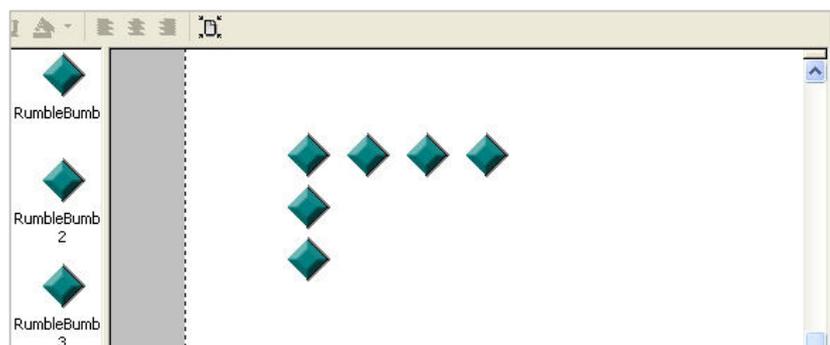


I guess that you can clearly see where I'm going with this...Changing ANYTHING in ANY of the duplicates applies to all of the duplicates and the original as well! In other words: you don't have four different objects here, you only have four different occurrences of a single object! That's really pretty important, so try to keep this in mind whenever you'd like to copy an object.

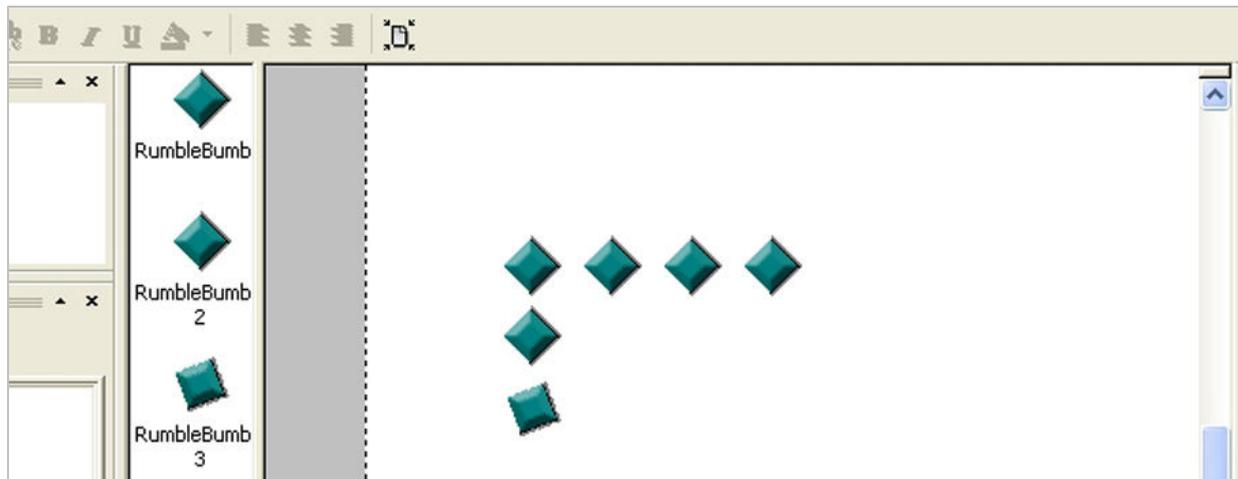
Now, let's get our objects to the previous state. Select the "Undo" command from the "Edit" menu or just use the **Ctrl+Z** keys.

It's time for some **cloning** now. Right-click on the first object to the left (it doesn't really matter on which one you click, I'm just trying to keep some order here, so that we'll have the same results) and select the "Clone Object" command. The window that appeared looks kinda' familiar, doesn't it? Sure, it looks exactly the same as the duplicating one. Remember what we said earlier? We wanted TWO clones. So, set the rows number to 3, leave the columns at 1, set the row spacing to 5, and leave the column spacing at 0. Click OK.

Take a look at the image to the right – that's what I received. Notice that there are new icons on the **Objects List** – there's not only RumbleBumb, but also RumbleBumb 2 & 3!



Now, let's do another experiment. Select the **RumbleBumble 3** object and change its size or angle (if you don't know which one of the objects is RumbleBumble 3, select it from the Objects List or hold your mouse pointer over the objects one by one, for about 2 seconds – a tooltip will appear, containing the name of the object that's currently under your cursor). Got it? Great! I guess that your results are similar to mine:



As we can see here, changing anything in a cloned copy doesn't change the second clone, nor the original object. That's because cloning creates a completely stand-alone object, not just an occurrence of the original. The point is: use duplicating when you want to create a lot of identical objects behaving in a common way, sharing attributes with other objects of the same kind (for example: a pack of enemies that act similarly to each other, bounce around and shoot the player with a laser whenever they see him), use cloning when you want to create unique objects (for example: an enemy that throws knives at the player whenever he sees him and an enemy that tries to run away whenever the player shoots his rifle).

Additional hints:

- Using **Alterable Values** will help you to overcome any problems with duplicates that you might encounter – but that's a topic for another tutorial, since it's a bit more advanced technique (hint: if you're really interested, take a look at the *Glob Wars* tutorial – you can find a thing or two about Alterables there).
- Duplicated objects usually take less memory than cloned ones (as they all have shared properties).

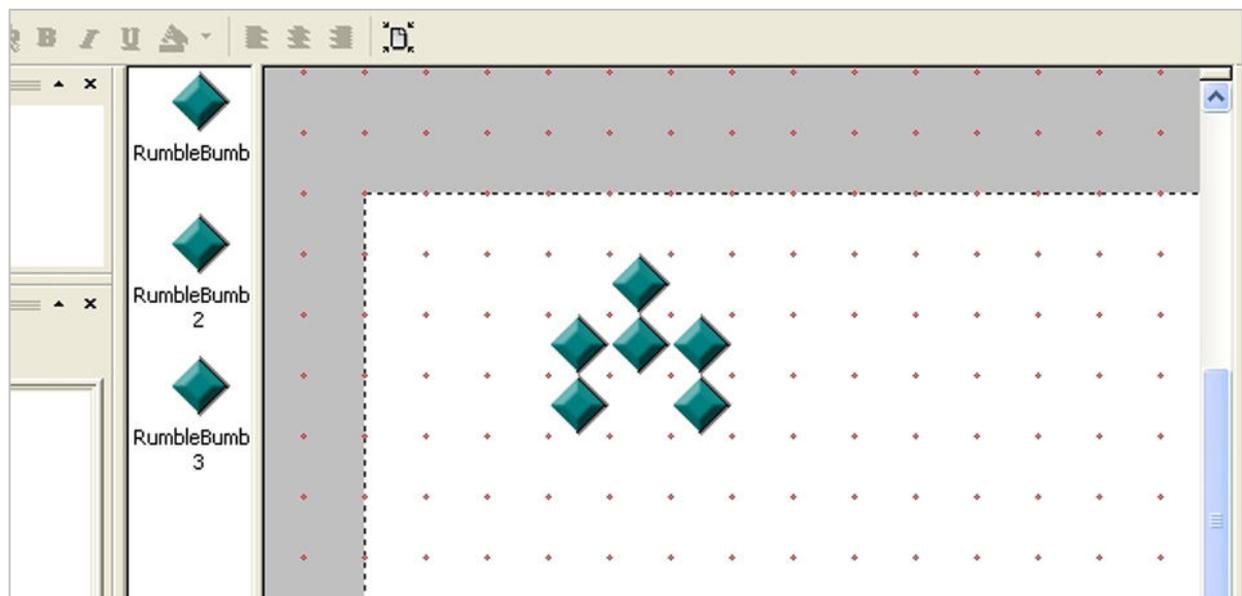
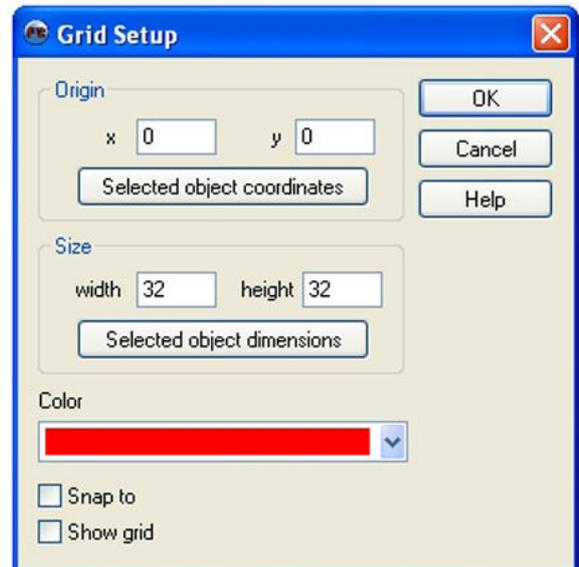
So, we now know how to easily rotate, resize, clone and duplicate our little RumbleBumble object...Yet, somehow we managed to skip one of the most basic things ever... We still don't know how to move our objects around! It's horridly simple, actually, and pretty self-explanatory, but let's just say a thing or two about it, just to be sure that everyone will get the point. The easiest way to **move objects** around is by dragging & dropping. Just drag an object to the desired destination and release the mouse button. A bit more complicated way (but still pretty simple) is by editing the properties of an object (in the **Properties Toolbar**) – just change the **x** and **y positions** of an object by entering different values there, and your object will automatically jump to the given coordinates. The third way – the most precise, and often used to reposition an object just by a couple of pixels – is to use arrow keys on your keyboard. To do so, select an object and press one of the arrow keys to move the selected element by one pixel in the given direction.

Both the cursor keys method and simple dragging & dropping also work for moving multiple objects. To **select multiple objects**, either multiselect them with your mouse (drag a selection field over them) or click on each one of them while holding the **Shift** key. To select ALL objects in the frame, simply press **Ctrl+A** on your keyboard, or go to the “Edit” menu and select the “Select All” command. In the same menu you can also find another quite useful command: “Invert Selection”.

Additional hints:

- Take a peak at the “Arrange” menu – you can find quite useful tools there too, such as **locking and unlocking** objects, **changing the alignment** of objects, **flipping** them, **spacing** them or **making them the same size**.
- To change the **order of the objects** (z-position), simply right-click on one of them and go to the “Order” sub-menu, where you can find commands such as “**To Front**”, “**To Back**”, “**Forward One**”, “**Backward One**” and the possibility to set the order by objects' x or y position (note: you can also find the “Order” category in the “Arrange” menu).
- Here's another method for quickly duplicating objects: When you drag an object within the Frame editor, just press and hold the **Ctrl** key – instead of just moving, the object will be duplicated.
- Press **Ctrl+B** to exit the Frame Editor and open the Storyboard Editor.

One more thing to talk about, before we wander off to the big ol' Event Editor. And that "one thing" is a very useful tool, that can save you a lot of time and nerves on some projects – the **grid**. Multimedia Fusion's 2 grid system is very easy to use - you just set it on, and then open the grid's setup window (take a look at the screenshot to the right), where you have to decide whether you'd like it to be a **snapping grid** or not (if not – then it can be used as a simple visual aid), decide what size (in pixels) should the grid be... And that's all! Well, not exactly, because you can also set the origin of the grid (where on the screen should the grid start) and it's color, but that's not really necessary, as the default settings are pretty efficient. Would you like to experiment a bit with MMF2's grid and it's settings? Let's do that! Go to the "View" menu at MMF2's main toolbar – click on the "**Snap to grid**" and "**Show grid**" buttons, and then enter grid's preferences, by clicking on the "**Grid setup**" button. Select the size of the grid to be 32x32 pixels (it should be set by default). Click OK. Now, try to move one of your objects by dragging it with the mouse... It works! This is what I've got after a few seconds of dragging & dropping:



When you're done playing with the grid, move on to the next part of this tutorial. We're going to have a short overview of the Event Editor... So... Carry on!

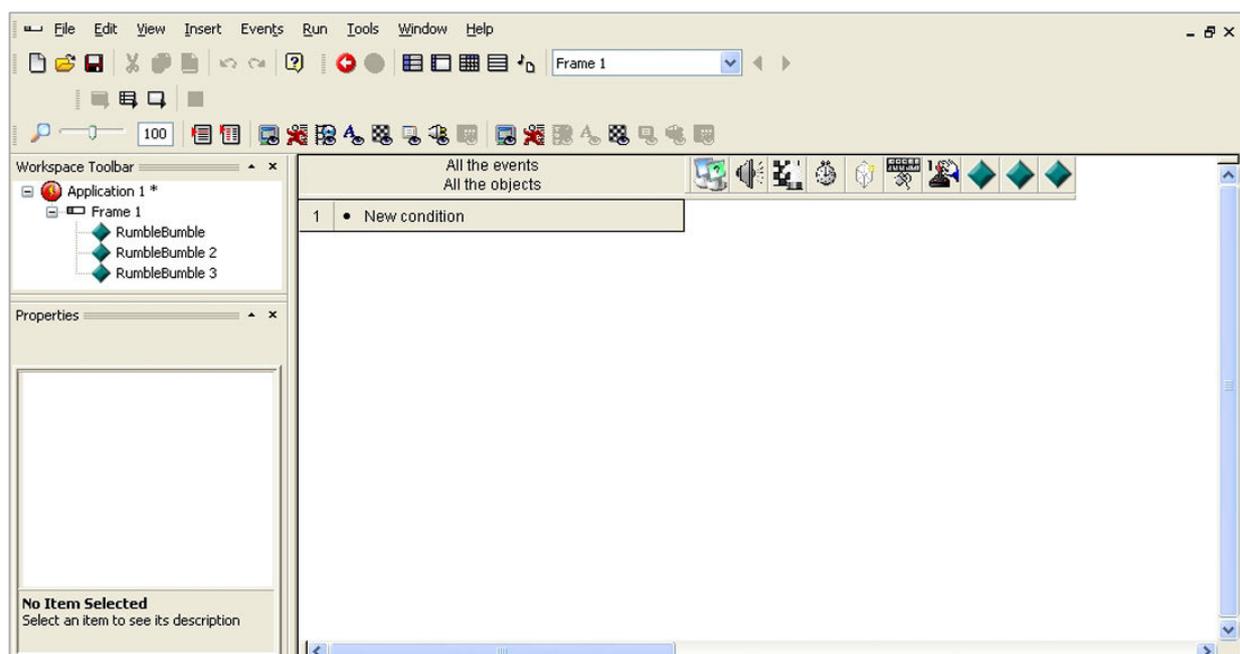
Part V: The Event Editor

The **Event Editor** can also be known as the “mind” or “control center” of all MMF2-created games and applications. This is the place where you create and control events (sets of conditions and actions), which are the basis of the computer logic in your projects. Without the Event Editor, MMF2 would be just a hollow shell. Thanks to EE’s power and ground-breaking design, MMF2 is considered to be the best and most efficient multimedia tool out there. Whether you’d like to play some music in your game, make one of the enemies move or control the health meter of the player – this is the place where you’ll end up in. Believe me – thanks to the Event Editor, you’ll soon find yourself creating some really neat and classy games & multimedia projects!

OK, we’ve got the introduction behind us. Now, let’s find a way how to even start the Event Editor. Well, nothing easier, actually - go to the top menu, select **View > Event Editor**. Or, you could just press **Ctrl+E** key combination on your keyboard. Or – there’s always the possibility to open the Frame Editor by using the **Navigation Toolbar**. To do so, just find your way to the button pointed out at this image:



Once your at the Event Editor, a similar (or identical) view will hit you right in your eyes:



Looks a bit strange, doesn't it? But don't worry. It's really pretty simple to handle. A list of conditions is displayed vertically to the left, whereas corresponding actions (which are initiated if the given condition is true) are listed to the right. And that's really the most important thing you need to know here. As I said: it's frighteningly simple!

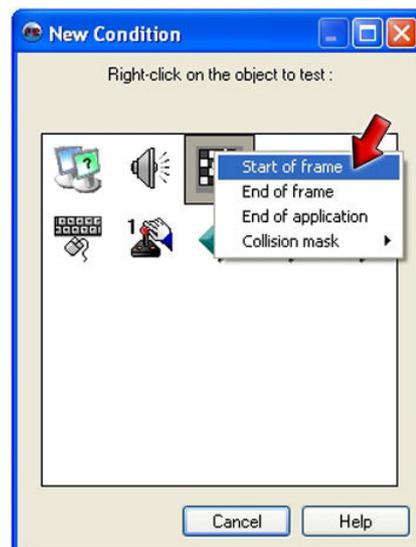
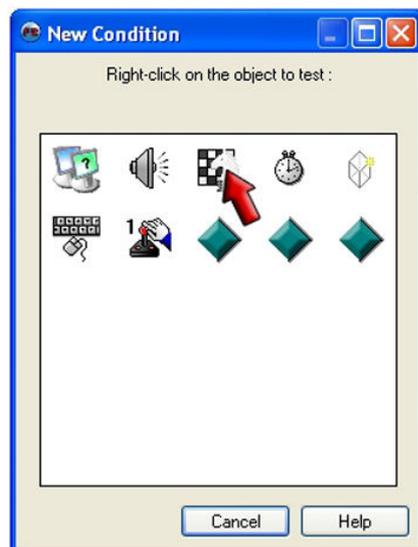
Let's see what the MMF2's help file has to say about this: *"The event editor displays the conditions and the actions in innovative way, like a spreadsheet. The conditions are listed vertically in the left column, and the action horizontally in the grid. To enter a new condition simply click on the New Condition line displayed in the left column, then right click on the object you want to test in the dialog box, and choose the condition in the popup menu. To enter a new action, locate the condition you want to affect with the action, and locate horizontally the object you want to modify. Click with the right mouse button in the square located under the object and in front of the condition: a popup menu appears and you only have to choose your action in the list".*

Still doesn't ring a bell? Let's do it step by step, then! All you have to do here, is to follow the given instructions and watch for the points-of-interest, pointed out by red arrows:

1) Click on the "New Condition" line:



2) A "New Condition" dialog window appears (take a look at the two images below). It shows a list of available objects (including the MMF2 logic ones). You need to right-click on an object to open a sub-menu containing available conditions, and then choose a condition

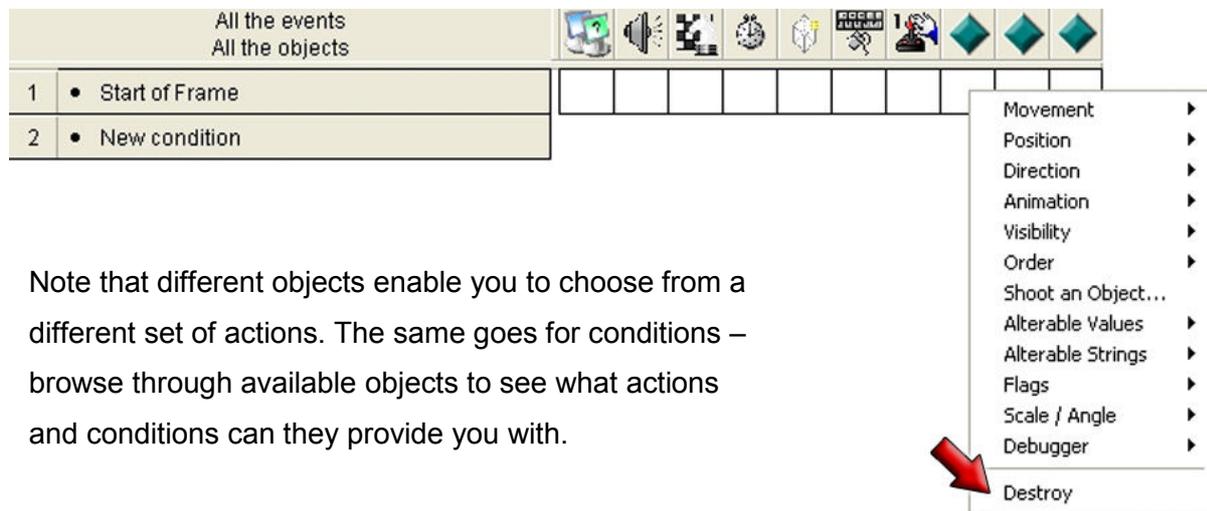


(simply follow the red arrows seen on the two images to the left to create a new condition that will be true only at the start of a frame – such a condition initiates the later given action(s) always when the frame starts).

3) We now have the “Start of Frame” condition set up. It’s time to add some actions to this event (actions that will be initiated if the given condition is true). Let’s say that we want to destroy the **RumbleBumble** object (all of it’s occurrences, including the duplicates) as soon, as the frame starts. So, since we already have the “Start of Frame” condition, it’s time to add the proper action. Find the RumbleBumble object on the horizontal list and right-click on the empty field corresponding to the “Start of Frame” condition line.

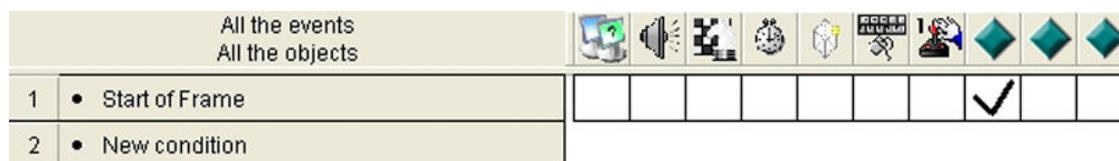


4) Select the “Destroy” action from the pop-up menu. As shown on the image below:



Note that different objects enable you to choose from a different set of actions. The same goes for conditions – browse through available objects to see what actions and conditions can they provide you with.

5) And that’s that! We have set up a fully working event!



Additional hint:

- The drag & drop technique works mighty fine in EE! Just check it out for yourself!

There's a lot advanced techniques, tools and tricks that can be found in the Event Editor – but don't worry about that yet. All in due time, cadet! If you'll spend some time playing with MMF2, discovering it's possibilities and creating some smaller projects for a good start, you'll soon know all the things you should know about loops, groups and optimizing your code for better performance. Just don't worry about that yet, and always remember – all in due time!

Additional hints:

- **MMF2's engine goes through the list of events one by one, starting from the top** – remember this when playing with your projects! The sequence of events can sometimes have a great significance for the whole code of your game or app!
- If you wish, you can **modify the order of the actions in a single event line** (for example: making one of the objects firstly shoot, and then be destroyed). To do this, double-click on a cell, or right-click the event number and select "Edit the actions" from the pop-up menu.
- There is an alternate Event Editor embedded into MMF2 – The **Event List Editor**. It is not as widely used as the normal Event Editor, and as such, isn't a part of this tutorial. But if you wish to explore it on your own someday – don't hesitate. Perhaps you'll find it to be more efficient for you, although I advise all newbies to stick to the standard Event Editor.

Part VI: Testing your projects

Once you've created a couple of events, it's always a good idea to test them on runtime (to run them as an application) – just to see whether they work in a way they are supposed to, or not. Testing apps in MMF2 is as easy, as it could be. You just push a single button and – voila! – you're already in the middle of a testing session. Firstly, take a look at the top of MMF2's interface. You'll find a small toolbar there, looking like this:



This is the **Run Toolbar**. It enables you to quickly test your whole project, application or just a

single (currently selected) frame. Just a single click – and all those little cogs start to spin, your application comes to life! You can access the commands from this Toolbar through the means of the **Run** top menu or keyboard shortcuts – press **F9** to test the whole project (series of applications), **F8** to run the current application, **F7** to test the currently selected frame.

As soon as the test run initiates, a special toolbar appears on the screen (take a look at the screenshot below), enabling you to restart, stop or pause the test. It also displays a bit of information about your current app and frame (such as the number of active objects), more being available after pushing the “+” button.



Part VII: Koobare’s MMF-to-paper coding system

Smile up, cadet, you’re almost at the end of this here tutorial! You currently know a thing or two about the main interface of MMF2, about it’s editors and their usage... But there’s still a thing that you should learn before going out on the fascinating trip through the land of creativity. What’s that? – you ask – And why should I spend my time taking a look at it?

Because, my dear sith apprentice, this can be a bit useful when you’d like to show off some new coding techniques that you’ve made, or when you’ll want someone at the forums to help you with your code...And you’re surely going to need this when you’ll decide to take on one of my other tutorials!

Whether you find yourself interested, or not... Let me introduce you to my MMF2-to-paper event-recording system! A simple way to scribe down all those MMF2 events on your sheet of paper or some text processing program! The whole thing looks like this:

IF (Condition): **[Object for the condition]** > **Condition group** > ***Condition***

THEN (Action): **[Object for the action]** > **Action group** > ***Action***

All the conditions are marked in red color, while actions are written in fancy blue. Object names are always put in [square brackets]. The final condition/action is always in *Italic*. If we’ll have a multi-condition event, then we’ll have:

IF (Condition 1): [Object for condition 1] > Condition group 1 > Condition 1

IF (Condition 2): [Object for condition 2] > Condition group 2 > Condition 2

THEN (Action): [Object for the action] > Action group > Action

Whereas a multi-action event looks like this:

IF (Condition): [Object for condition] > Condition group > Condition

THEN (Action 1): [Object for the action 1] > Action group 1 > Action 1

THEN (Action 2): [Object for the action 2] > Action group 2 > Action 2

THEN (Action 3): [Object for the action 3] > Action group 3 > Action 3

If you'll have to input anything by keyboard (for example: a value to set the counter to, or a text that is going to be displayed with the alterable string option – or other things that you use the expression editor for) it will be indicated by coloring the text in green and using < angle brackets >, like in this example (note that sometimes the given text will be set *Italic* for easier recognition):

< Set the Global Value A to 32 >

Additional comments, info and instructions will be put in << double angle brackets >>, using a different color:

<< Select any wave sound from the MMF2's sound library >>

Seems pretty simple, right? Because it is! You'll see – it works as a charm!

And here's the grand finale!

And that concludes our little tutorial! Thanks for your time, I hope that you learned something interesting today. Some aspects of the MMF2's interface and usage weren't covered in this tutorial, so you can expect to find another one – this time about the Image Editor or more advanced aspects of the Event Editor – soon enough! Meanwhile, try to experiment on your own, create some neat games and applications (I'd suggest starting with smaller projects at the beginning), check the educational resources on Clickteam's website... MMF2 is a powerful (and I really mean POWERFUL!) program and there are always lots of tricks and techniques

to learn! Run through your MMF2 manual, download other tutorials, join the community and don't hesitate to ask about anything! And always remember: **practice makes perfect!**

Thanks for your time and see you again soon!

Cheers!

Koobare
marchewkowy@gmail.com

*If you have any questions, suggestions or just need help –
mail me at marchewkowy@gmail.com*

If you have any problems with this tutorial, or notice that there are some mistakes present, please, contact me and I'll do my best to help you and replace all the errors with correct information.

Contact me at: marchewkowy@gmail.com