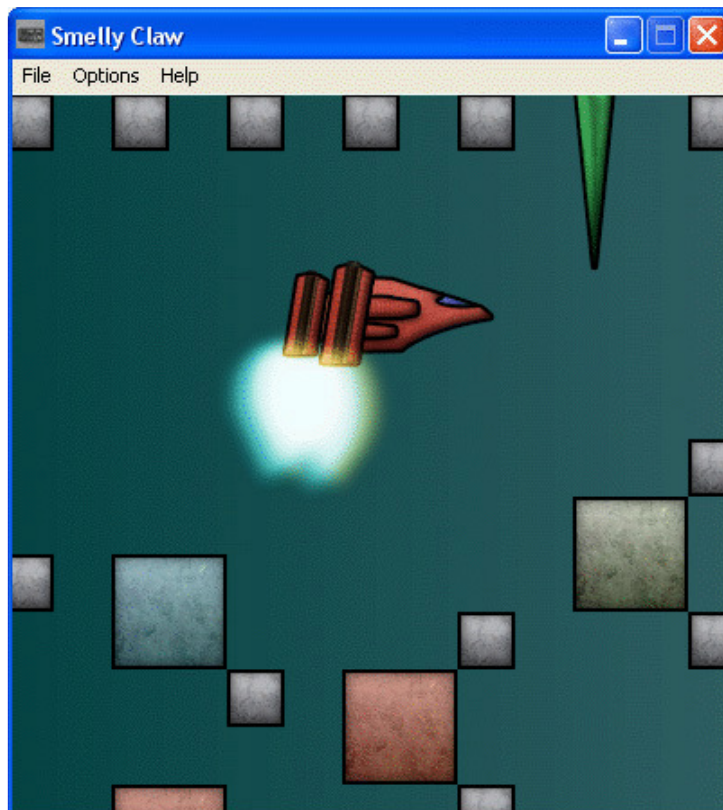


# the **SMELLY CLAW**

tutorial  
For Multimedia Fusion 2



You may not use this tutorial for any other purpose than learning, working and 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.

**Koobare**  
marchewkowy@gmail.com

## Hi there, all!

---

Welcome to Koobare's pretty little tutorial, explaining how to create a "Smelly Claw" game using the best multimedia authoring tool ever – [Multimedia Fusion 2](#) by Clickteam!

Don't know what's the "Smelly Claw" all about? Don't worry, You'll soon discover that. Basically, it involves a small starship (named "the Smelly Claw" – and thus the title becomes pretty obvious... And no, it does not have anything to do with X-Men's Wolverine or a Yeti) flying around and crashing into stuff, trying to find it's way through a maze of cosmic blocks, blades and other weird thingies like that. The basic idea for this game is pretty simple, adding just a bit of a twist when it comes to steering... But let's leave that for Part V of this here tutorial, shall we?

### Here are some basic features of the game we're going to create:

- The objective of the game is to cross a maze of blocks and moving blades, without colliding with any of them.
- Player's spacecraft flies around using a basic jet-booster system: it has to be switched *on* for the player to maneuver the ship (fly to the right or left). As soon as it's *off*, the Claw just starts to fall. Player has to move around using his ship's jet engine and free-falling to reach lower-situated passes.
- When Smelly's booster is on, a groovy graphical effect for the engine's blast is displayed – using the "ADD" ink effect and an alpha-channeled image.
- If the Smelly Claw crashes – the game places it back at the beginning of the level and restarts the frame.
- Game utilizes a basic scrolling system.
- Player's movement is based on a simple trick using the Bouncing Ball movement.
- Game is displayed in a 400x400 px window, without the possibility to maximize or resize it.

### Part I: Setting up the application

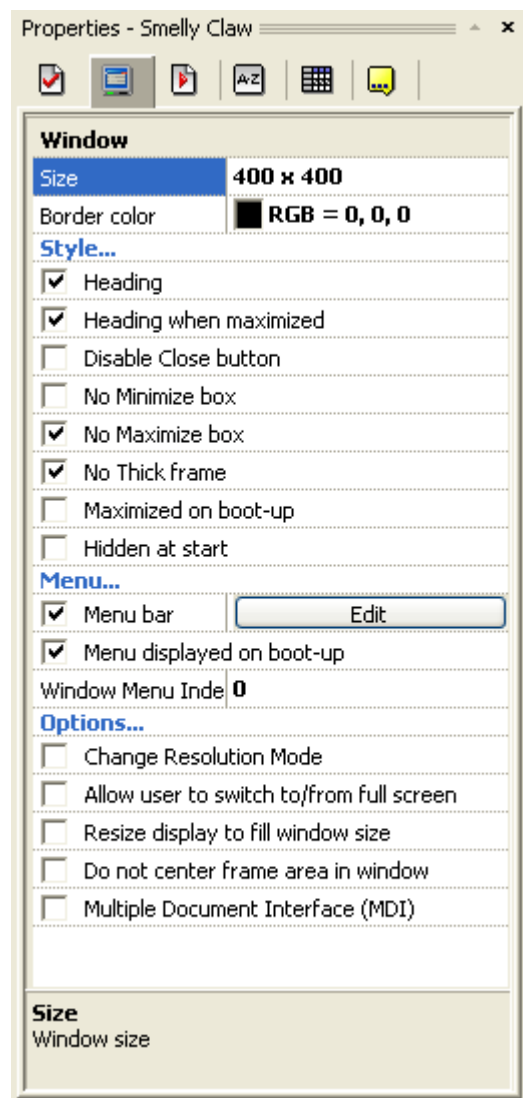
---

So, You think that You're good enough to create the "Smelly Claw" game, do You? Great! That's the spirit! Show me Your scary face! Come on, no one's looking! Ahhh, that's what I wanted to see! A real blood thirsty click-warrior, ready to create a game of his lifetime!

Now, let's get on with it, shall we?

**Create a new application in Multimedia Fusion 2.** Name it, save it, keep it. Now, let's go to the **properties screen** (if it didn't open by itself, right click on Your app's name in the *Workspace toolbar* and select "*Properties*" from the drop down menu). Set the *size* of the game's window to **400x400 pixels**.

This will enable us to have a bit smoother & faster scrolling in our game – scrolling the game’s frame always eats away quite a lot of our system resources (especially on older computers), so it’s good to keep the window a bit smaller than the usual 640x480. Your RAM memory will thank You for this someday. Flowers, champagne, a nice evening at the movies. And it’s all just for keeping that window size on a smaller level.



Now, remember what it says in the *features list*? This game cannot be resized or maximized by the player. I want it to keep the same size it starts with, and I don’t want to see any maximized windows out there. Check the “**No Maximize box**” option. This will make the maximize box unavailable (grayed out). Then, click the “**No Thick frame**” option too - the application window will be drawn with a thin line instead of a thick one, which means that the game’s window won’t be resizable.

Setting up properties in MMF2 is a piece of cake, isn’t it? And I mean a bit of a chocolate one! With some crispy, scrumptious caramel on top...Yum-yum.

Check the screenshot to the left if You feel lost – we should have just made our properties screen look something like that. Does it look like this?

If You said “yes” (and I really hope You did) - let’s move on to the next part.

## Part II: Setting up the frame, importing objects.

**Firstly, let’s set up our frame.** We already know that we want our game to be a scrolling one – in other words: we need to make the frame size **bigger** than our window size, so that the window can show only a part of the frame at once, and has to scroll through it to show the rest. Pretty simple, eh? Remember how big our window size was? (Of course You do, it wasn’t a year ago, was it?) Let’s set the frame size bigger than those 400x400 pixels then. Let’s set it to **1024x768**. Just select Your blank frame and either use those size-changing shortcuts that You have under the frame’s title, or go to frame’s properties and change the frame size there. Ready? Well then, what are You waiting for? Move in! Open the frame!

Hmmm... Kinda' empty here, isn't it? Kinda' white too. Looks like the Matrix has been formatted. Bye, Keanu. See ya, Morpheus. Well, You guys weren't my all-time favorites anyway, so just good riddance. But this whole white, unused space is waiting for some objects... So let's not keep it waiting any longer.

I have created some nice objects for You to use in Your game.\* You just need to **import them into Your project** (TGF2 users – read below). The easiest method of doing that is to open the application that contains those objects, copy all of them (using CTRL+C combination, or the command from the “Edit” menu) and paste them into Your app. Let's do it now. Open the “**smellylibrary**” file (*smellylibrary.mfa* – don't open the *smellyclaw.mfa* file, that's not the object library, that's the tutorial example!), which was packed into the same archive as this .PDF tutorial. Select all of the objects (either by dragging the selection field with Your mouse, or by clicking on each one of them while the SHIFT key is pressed), press CTRL+C, go back to Your application, open Your empty frame and press CTRL+V. Click somewhere in the middle of Your frame and - Hooray! – You've just obtained all the needed objects. You can close the “smellylibrary” application now, or You can keep it open, it's Your choice.

**Let's take a closer look at those objects, shall we?**



**The Smelly Claw** - Player's starship. The pride of the Imperial Navy! The fastest ship in the Galaxy! The... Yeah, right. Just an old starship without any armor, weapons or any special features. A flying tin can, to be exact.



**The Murderous Blocks** - Worst enemies ever seen in a computer game. Hideous monsters, waiting to feast on human flesh. The Murderous Blocks come in four versions: three big ones and a smaller one. They all do the same. Be afraid of these foes! Beware! Beware I tell ya'!



**Spikey Spike (aka The Moving Blade)** - No, Spikey Spike is not a rapper. It's a moving enemy. Yep – thanks to my superior Artificial Intelligence scripting skills, Spikey can even move! I use a special AI system known as the “Path Movement” to do that. You'll find more info on the way.



**The Exit** - Seen in thousands of places, the Exit sign helps us find exits since 1000 B.C. It has an extra special feature in our game – It's not only pointing to the exit, the Exit sign IS the EXIT. And that's the way it should always be.



**The Quick Backdrop** - No one knows what this is. It spreads quickly, though. And - most probably - is just a backdrop.



**The Mighty Booster** - What would You expect from a booster that comes from a ship named “Smelly Claw”? No, it does not stink! Well... Maybe it doesn’t smell to good either, but “stink” is such a nasty word. Anyways – take a good look at this little stinker, since You’re gonna’ see it everytime the “up” button will be pushed.

OK... Now that we all know each other a bit better, let’s just set up all the preferences of these little objects and we can go to the part where the fun begins!

\* **These objects are using the Alpha Channel feature that is not available in TGF2!** TGF2-clickers should use standard library objects instead of the ones that I’ve created, or wait for a modified version of this tutorial, suitable for TGF2.

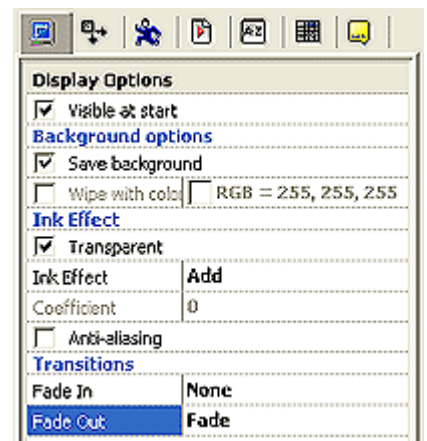
### Part III: Setting up objects preferences and movements.

---

Before we start to create our own level, let’s set up the preferences of those objects that could use a little tune-up. Don’t be worried if something stays unclear at this phase of our work – it will, most probably, have more sense when we will get to the coding.

- 1) **Select the Quick Backdrop object.** Make sure that it’s “Obstacle Type” setting is set to “None” (that way it isn’t counted as an obstacle by our program and does not interfere with player’s movement). If You don’t like the gradient colors that I’ve picked – pick Your own colors.

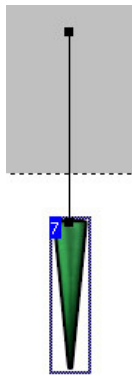
- 2) **Select the Mighty Booster.** Go to it’s preferences. **Set the ink effect to “ADD”** – this will make the object brighter (if it’s laying on a frame’s white background it can be hard to see – You may want to drag it out of the frame, and put it right by it, on the grey workspace, or change the frames background color) and will help us to create a cool booster effect. **Set the FADE OUT (NOT the “fade in”!) transition to FADE** and set it’s timer to 0.11 seconds (this value can be a bit higher, but this could result in some minor slowdowns on older computers). Take a look at the image to the right – that’s exactly what we want to achieve.



3) **Select one of the Murderous Blocks.**

Make sure that they all (all four) have the “Obstacle Type” set to “Obstacle”. That way we can – with a help of a single event – make all of them, duplicates included, a threat to the Smelly Claw.

4) **Select the Moving Blade (Spikey Spike).** He'll need some movement, right? So, let's set it up right now. Go to the Moving Blade's properties, select the third tab from the left (the little man running), open the dropdown menu and select the “**Path**” movement. Now, click on the “Edit” button. **The Path Movement toolbar** should appear:



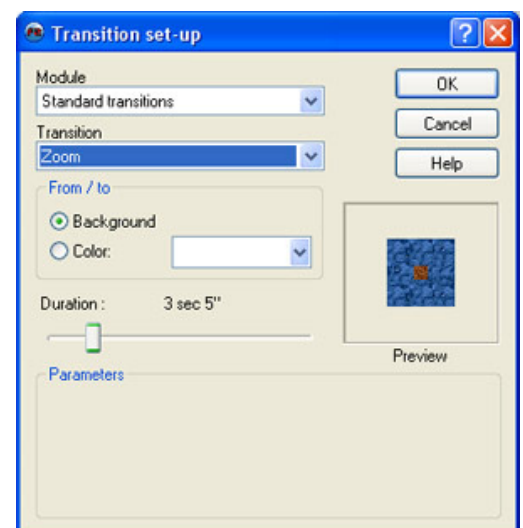
Using the first button to the left (the one with a single line linking a box and an arrow) select the “**New line**” tool. Click about 150-250 pixels above the Moving Blade to create a path similar to the one shown on the image to the left.

Now, select the first node of the movement (click on the first of the two rectangles that have been created and linked with a path), set the speed – using the speed slider – to 20, or 25, something like that. Do the same thing for the second rectangle too. Push the “**Loop the movement**” and “**Reverse at end**” buttons (those are the fourth and fifth buttons counting from the left) to receive a nice looping path movement. Click OK. Now, wherever You'll put the Moving Blade – or it's duplicates – it will move via a path, up and down, up and down, up and down... With a help of a single event we will make sure that when the player collides with the Moving Blade – it will destroy his ship. That way we've just created another dreadful enemy, of which the player should be afraid.

5) **Select the Smelly Claw (player's starship).** At the *Display Options* screen in the preferences (the first tab to the left) set the **Fade In** to **ZOOM** and set it to 3,5 seconds, as shown on the image to the right.

Got it? Great! Go to the **Fade Out** effect then, select **ZOOM** and set it to 2.0 seconds. Click OK.

Now, let's set up the needed movement... Go to the Movement tab, and select the “**Bouncing Ball**”





**movement** from the list. Don't be worried – Smelly's not going to bounce like a ball, we select this movement just because it's the easiest movement to control from the Event editor, when it comes to built-in movements. In the Bouncing Ball movement properties, set Speed to 30 and Deceleration to 0.

**Create a counter.** Last thing to do here: create a new Counter Object, set it's minimal value to 0, and both maximal and current values to 2. **This counter will help us determine whether Smelly Claw is fully functional (if the counter's value is 2), destroyed (counter's value = 0) or is currently being destroyed (counter's value = 1).** The reason why we need 3 starship states instead of two (functional/destroyed) is pretty simple - I would just like to have our starship bounce of a wall one more time before vaporizing, after it's been hit. Let this illustrate my little plan:

- Player accidentally crashes into one of the blocks. 1 is subtracted from the counter (counter is now set to "1"), and the starship is bounced off.
- If counter = 1, then: don't allow the player to control the starship's movement.
- The starship – out of control – bounces onto another block. 1 is subtracted from the counter again (counter is now set to "0").
- If counter = 0, then: destroy the starship.

Easy, isn't it? :) OK, we're done with setting up all those preferences! It's time to create a level, and – afterwards - script our game in Multimedia Fusion's ingenious Event editor! Hooray!

## Part IV: Editing the level.

---

Here comes the fun part. Your objective is to create a fully functional level, using objects that we set up earlier. You have to choose a starting position for the player (just drop his starship at the place You want him to start at), create multiple Spikey Spike's and Murderous Blocks, set up the Exit and set up a background gradient for the backdrop.

Let's do the last thing first – just select Your **Quick Backdrop**, move it to left-top corner of the frame, grab it's resizing points with Your mouse (those little rectangles on the QB's sides that appear when You double click on it) and then just cover the whole frame with it, by resizing it until it fits the playframe perfectly. When it's done, just double-check if Your Quick Backdrop is at the back (right-click on it and select the "To Back" command from the "Order" menu – if it's unavailable, then Your QB is already at the back). Ok, we've got the background, now it's time for those Murderous Blocks!

Multimedia Fusion 2 has a little tool that could help you a bit with those blocks, you know? They are either 64x64 or 32x32 pixels, so if we could set up a snapping grid, that would help us position those

objects in the frame, it would be cool. 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. Click OK. Notice the grid all over your playfield (don't worry, it's just in the editor – players won't see it in the game!), try to move one of your objects by dragging it with the mouse... It works!

You can now play a little and set up any level that you'd like to. Just remember about a few rules:

- a) Since the Smelly Claw will crash when colliding with an obstacle, be sure to give it some space to maneuver! Impassable levels aren't really that fun!
- b) It's best to leave some free space underneath the place where You'll put the Smelly Claw at the start. Most probably the player will need a second or two to throw himself into the game's world. It would be a pity if the Smelly Claw would be destroyed because of those two seconds.
- c) Limit the playground – make sure that the player's ship can't fly away from it (this could be limited via an event in the Event editor, but I think that it gives a better feel if those Murderous Blocks are watching You from every corner).
- d) Don't clone backdrop objects – duplicate them, use the CTRL+C and CTRL+V keys or just drag & drop objects from the object selector.
- e) If You need any help or inspiration – just take a look at my "Smelly Claw" tutorial example – open the *smellyclaw.mfa* and check out how I built that level).

When you're done – save your game! I hope that you saved earlier, if not – You're a pretty brave person. :) Remember to always save your game often and keep some backup copies of it!

After saving, proceed to Part V.

## **Part V: Time to rock. Playing with the code.**

---

Now, let's get our game moving with a few simple commands! Go to the **Event editor**. Let's get it all moving and working in the right order!

### **Player movement**

Remember what I have told You about player's movement? Let me refresh your memory: "*Player's spacecraft flies around using a basic jet-booster system: it has to be switched on for the player to maneuver the ship (fly to right of left). As soon as it's off, the Claw just starts to fall. Player has to move around using his ship's booster (which is switched on as long as the player holds the "up cursor" key) and free-falling to reach lower-situated passes*".

Let's do this!



a) Firstly let's set up this condition: **Repeat while "Up Arrow" is pressed**. Just create a new condition, go to the Keyboard & Mouse object, right-click on it, go to the "Keyboard" condition group and select "Repeat while key is pressed". MMF2 will ask you to input which key do you have in mind. Press the **UP** arrow key. That's done!

Let's expand this condition a bit. **Right-click on it and select the "Insert" command** (that's the way to create conditions set up from multiple sub-conditions, like when You want something to happen only when "A" AND "B" are both true). Select the counter object that we've created in part III (the one that will measure Smelly's condition) and choose "**Compare the counter to a value**" from it's action list. Select this comparison: Is equal 2. This means that this whole condition will be true only if the up arrow key is pressed and the counter's current value is equal 2. So, if the starship is destroyed or it's destruction is imminent (and thus the counter is not equal 2), then the player loses his control over the ship.

Ok, let's add the last thing to this condition – do you remember that we gave a fade in transition for the player's ship? It was a ZOOM fade in, set to 3,5 seconds. That means that every time this frame is restarted, Smelly Claw will zoom in onto the playground, taking 3,5 seconds before the player can move him, start his engines, etc. So, let's limit this condition so that it **is true only when the frame's timer is greater then 3,5 seconds** (or something like that). This isn't really necessary for the game to work fine, but will be useful when we'll add actions that create those Mighty Boosters near old Smelly's engines (this way they won't be shown until Smelly's zooming in ends). Insert another condition to this event then: right-click on the Timer object, select the "**Is the timer greater than a certain value?**" command and input the desired value (I've selected "03,40 seconds").

We should have established such a condition:

- Repeat while "Up Arrow" is pressed
- Timer is greater than 03"-40
- **2** = 2

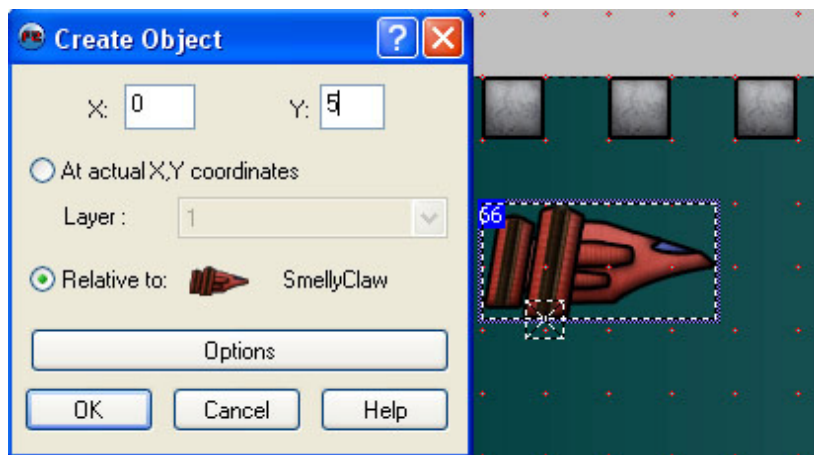
If Your sequence of subconditions is different, just move them up and down (by dragging them with Your mouse) until the "**Repeat while Up Arrow is pressed**" is at the first position – it's not the time and place to explain why, but it's always best to have "check for keyboard/mouse input" subconditions at the top.

Now, it's time to add some **actions** into this event. Go to the Smelly Claw column, right-click in the white field that's corresponding to the row with your newly-created condition. Go to the "Direction" action group and select the "Select Direction" command. A dialogue box will open – let's pause here for a moment. Just take a look at it. It looks like a big circle made of rectangles. These are the directions that You can choose. Take a look at the way they are numbered – the "right" direction is represented by the number 0, the "top" direction – by the number 8, the "left" direction – by 16. There are 32 directions to choose from.

That means that if I wanted to select direction number 15, I'd choose the rectangle that is one space above the "left" direction.

Now, unselect all previously selected directions and select the one that leads directly up (click on the rectangle positioned most to the top – by the number 8). Click OK. Now, thanks to our deeds, every time the specified condition is true, Smelly Claw will select the direction "up".

Go to the "Create new objects" column (should be a bit to the left), right-click on the corresponding white tile and select "Create object". Select the Mighty Booster object and click OK. Then, using the dialogue box that allows you to select when would you like your new object to be created, select the 0, 5



coordinates (X=0, Y=5), relative to the Smelly Claw. You can take a look at how it should be done at the image to the left. Click OK and then repeat this action (just do the same things again, without deleting the previous action), but this time set the coordinates to -24, 0 (X= -24, Y=0), relatively from the Smelly Claw. This should be fine.

Thanks to the action that we've just coded, as soon as the specified condition is true, the game will create booster objects by the Smelly Claw's engine shafts.

**b)** Now, create an **"Always" condition** (right-click on the Special Object and select "Always") and set these actions to correspond with it:

- Right click on corresponding tile under the Storyboard Controls object. Select the "Scrollings" category and then the **"Center window position in frame"**. Select the position to be 18, -24 (X=18, Y= -24) relatively to the Smelly Claw. This will center the window to the position of the starship (right in the middle).
- Right click in the right tile of the Mighty Booster object. Select "Destroy" from it's actions menu.

What we've just done: on every program's loop, all created Mighty Booster's will be destroyed (don't worry about it – previously we've set an event that creates them on the same speed!), allowing us to see it's short – yet beautiful – fade out animation. Thanks to this – and the fact that we've added the "ADD" ink effect to all the boosters – on every loop that the player holds the "up" key, nice-looking engine's fire will be created right by the Smelly Claw. Plus, **we've got a working scrolling system**, thanks to the "center window position in frame" command!

c) Again, create a new condition – once again create the **“Repeat while Up Arrow pressed”**. Right click on the newly created condition and select **“negate”**. That means that this condition will only be true if the Up Arrow **IS NOT** currently pressed. Go to the Smelly Claw object and select the **“Select Direction”** action from the **“Direction”** category. Set it to the 24<sup>th</sup> direction (down).

You can check your application now, if you’d like to. Kinda’ cool, isn’t it? :) Watch how the Booster effects shine under your starship’s engines, thanks to the **“ADD”** ink effect and the fade out animation.

After you’re done admiring, let’s get back to work!

d) Create a new condition – **“Repeat while Right arrow is pressed”** (this time it’s the **RIGHT** arrow, not the “up” one!), and then insert another one: right-click on the Smelly Claw and select the **“compare direction of the Smelly Claw”** from the **“direction”** sub-menu (Select the **“UP”** direction). In other words: we’ve just created another condition that’s built of two sub-conditions. Now, create the **“Select direction”** action for the Smelly Claw object and set it to the direction number 1 (click on the directional box higher than the one fully to the right).

e) Create another event identical to the “d)” one, but with the **“Repeat while Left arrow is pressed”** condition instead of the **“Repeat while Right arrow is pressed”** and select the **15<sup>th</sup> direction** (above the one to the left) instead of the direction number 1.

f) Create a **“Smelly Claw collides with background”** condition (Select the Smelly Claw, select the **“Collisions”** menu and then the **“Backdrop”** option). When it happens, set the direction of the Smelly Claw to the **24<sup>th</sup> direction** (down), bounce it (select the **“Bounce”** command from the **“Movement”** sub-menu) and subtract 1 from the counter that’s checking Smelly’s condition (**“Subtract from Counter”** action).

In other words: when our starship collides with background obstacles, it’s being directed down, bounced and the counter changes to a lower value.

g) Create a **“Smelly Claw collides with the Moving Blade”** condition (Select the Smelly Claw, select the **“Collisions”** menu, then **“Another object”**, then the **“Moving Blade”** object) + add a sub-condition: **“Only one action when event loops”** (it’s under *Special Object > Limit Conditions > Only one action when event loops*). Then, create such actions: **[Smelly Claw] > Select Direction > 24<sup>th</sup> (Down)** and **[Counter] > Subtract 1 from counter**.

h) Create conditions: if **“Counter’s value = 1”** + **“Only one action when event loops”**, then play a sample (You can pick one from the MMF2 bonus CD – there are plenty to go around!), and set Smelly’s ink effect to **Monochrome** (pick Smelly Claw, select **“Visibility”** and **“Change ink effect”**, select

“Monochrome” from the drop-down list). This way Smelly will change his colors to black and grey ones when he is hit and out of player’s control.

i) Condition: if “**Counter’s value = 0**”, then **destroy Smelly Claw** (select “Destroy” from it’s action list). I guess that there’s not much to comment here. Move along!

j) Another event: create a condition by right clicking on the Smelly Claw and selecting “**Pick or Count**”, and then “**Have all [Smelly Claw] has been destroyed**”. Then, create action by the Storyboard Controls object: “**Restart the current frame**”. There’s no much to say here either. I could talk about donuts or transistor radio’s – but why the hell for? ;)

k) We’re almost over. Create another event, with these two conditions: [**SmellyClaw**] > **Collisions** > **Overlapping another object** > **Select [Exit Object]**, and if **counter’s value is 2**, then select the “Next frame” from the Storyboard Controls object. Of course, this has sense only when You’ll create a second frame to Your game.

l) And here’s the last event! Create it and reposition it to the top (for better organization): at “**Start of Frame**” (a condition selected from the *Storyboard Controls object*) play a sample from the MMF2’s library, play some MIDI music from the same source and set the counter to “2” (this isn’t necessary, but it can be useful if You forgot to set the right value to the counter in the Frame editor).

**And that’s that!** Check Your application, see if everything works! You can compare Your Event editor to the screenshot I made below (it doesn’t have to be identical, since You could have a different sequence of events or objects). Just take a peak:

All the events All the objects											2
1	• Start of Frame	✓									✓
2	• Always		✓							✓	
3	• Repeat while "Up Arrow" is pressed • Timer is greater than 02"-55 • <b>2</b> = 2				✓			✓			
4	•  Repeat while "Up Arrow" is pressed							✓			
5	• Repeat while "Right Arrow" is pressed •  is facing a direction							✓			
6	• Repeat while "Left Arrow" is pressed •  is facing a direction							✓			
7	•  collides with the background							✓			✓
8	• Collision between  and • Only one action when event loops							✓			✓
9	• <b>2</b> = 1 • Only one action when event loops	✓						✓			
10	• <b>2</b> = 0							✓			
11	•  is overlapping • <b>2</b> = 2		✓								
12	• Last  has been destroyed		✓								

Here's a bigger version, if your eyes aren't too well:

All the events All the objects											2
1	• Start of Frame		✓								✓
2	• Always			✓						✓	
3	• Repeat while "Up Arrow" is pressed • Timer is greater than 02"-55 • <b>2</b> = 2				✓		✓				
4	•  Repeat while "Up Arrow" is pressed						✓				
5	• Repeat while "Right Arrow" is pressed •  is facing a direction						✓				
6	• Repeat while "Left Arrow" is pressed •  is facing a direction						✓				
7	•  collides with the background						✓				✓
8	• Collision between  and • Only one action when event loops						✓				✓
9	• <b>2</b> = 1 • Only one action when event loops		✓				✓				
10	• <b>2</b> = 0						✓				
11	•  is overlapping • <b>2</b> = 2			✓							
12	• Last  has been destroyed			✓							

See? That was easy! :)

And now you should try to do it all again, but without this little tutorial to guide You. Maybe even try to expand the game's concept a bit? Add life counters, limited number of continues, bigger levels... Do whatever you'd wish to do! Remember: practice makes perfect!

So, you thought that you're good enough to create the "Smelly Claw" game, did you? Now we know that you were right!

Cheers!

**Koobare**

marchewkowy@gmail.com