

Pong tutorial - Step-by-step

How to make a game for 2 players in TGF2 / MMF2

Written by Popcorn

Welcome to the Pong tutorial. This step-by-step guide will teach you how to make a simple game in The Games Factory 2 and Multimedia Fusion 2. This is briefly what you will learn in this tutorial:

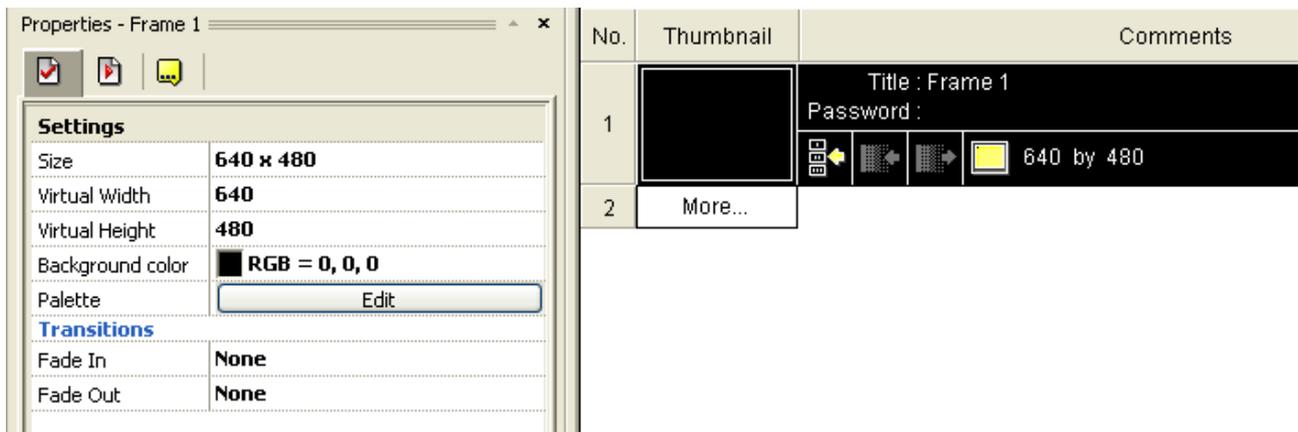
- How to set up the frame
- How to make events and actions
- How to assign movement to objects
- How to make a game for 2 players
- How to add lives in your game



This is how the game will be like

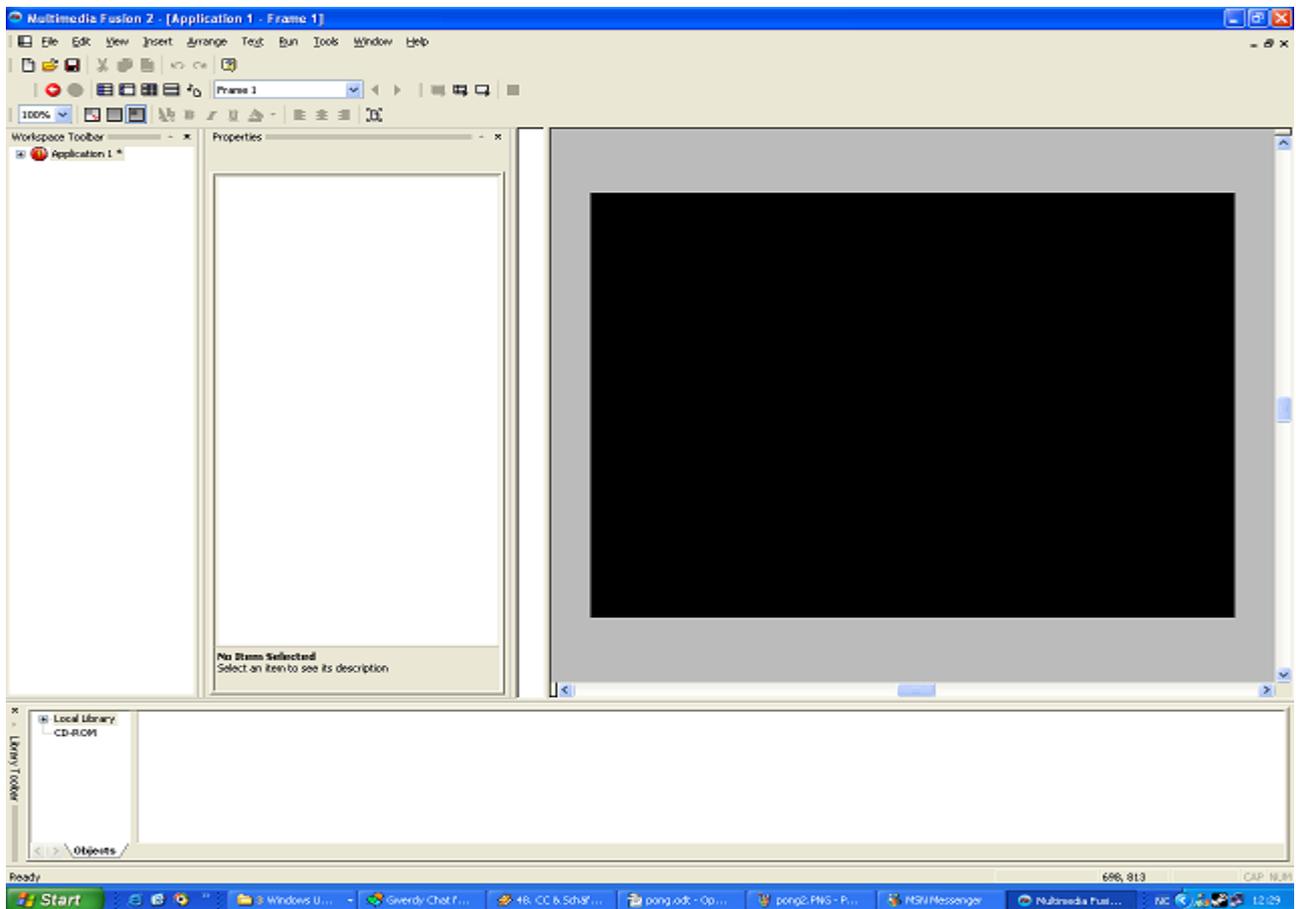
I assume that you have no knowledge of the product at all and I will therefore go as much into detail as possible. Since there are no differences between TGF2 and MMF2 for what I'll go through in this tutorial, I will talk about both the products, but only refer to them as TGF2.

Start TGF2 and select **New** from the **File** menu. What you see here is the Storyboard editor. At the right you should see a little white box. This represents a frame or level in your game. Click on it and you'll see that the properties for it will appear to the left.



As you can see I changed the background-color to black. You can set it to whatever color you want. Do not do any other changes now.

When you have selected a background-color, you are ready to enter your frame. Double click on the frame (that little box) or right click it and select Frame Editor. You should now be presented with an entirely black area (if you chose black as the background-color). That's the playarea.



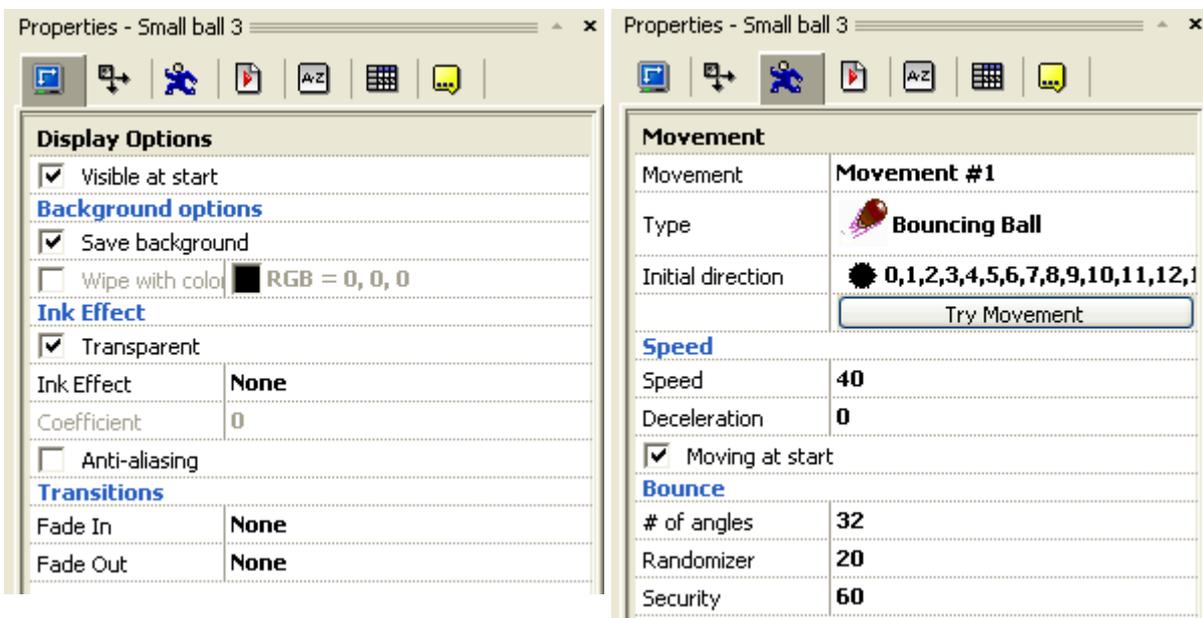
Do your screen look like the one above? You might have to resize some of the windows in order to see most of the play area as possible.

Ok, time to add some objects! The Library Toolbar should be visible at the bottom of the screen. If it is not, you have to select it from the Toolbars in the View menu. Here you can browse either the Local Library or the CD-ROM. If you have installed the gfx library to your TGF2 folder, all gfx can be found in the Local Library. If you haven't you have to find them from the CD-ROM. We want to find some nice gfx to use in our Pong game. Browse through the library and find Games / Miscellaneous. Then double click on Game Objects 2.



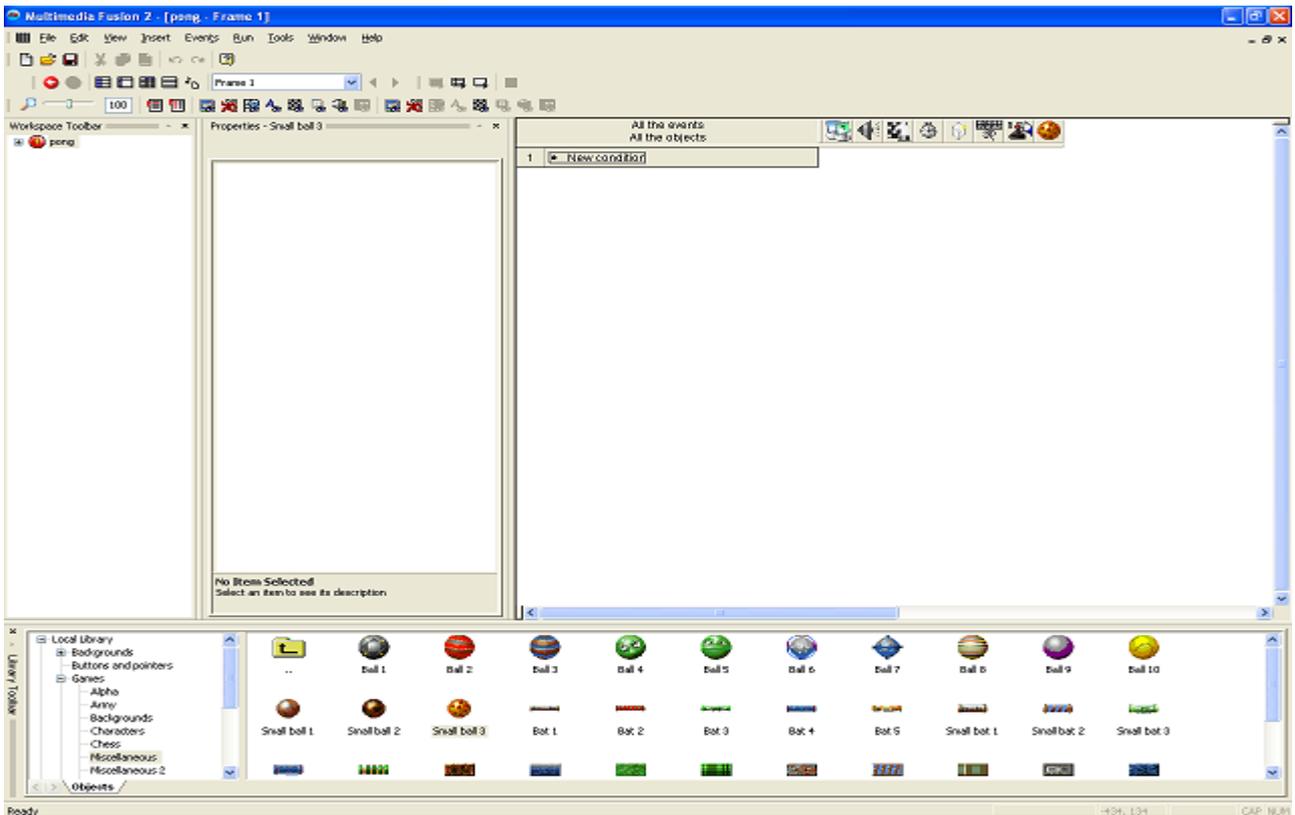
The first object we want to make is a ball. So find a good looking ball and drag it onto the playarea. I chose Small Ball 3, but you can choose whatever you want. It doesn't even need to look like a ball, but that's of course an advantage if it does. Leftclick on the ball. You should see it's properties in the properties window.

Select the Movement Tab (the one with the running man).

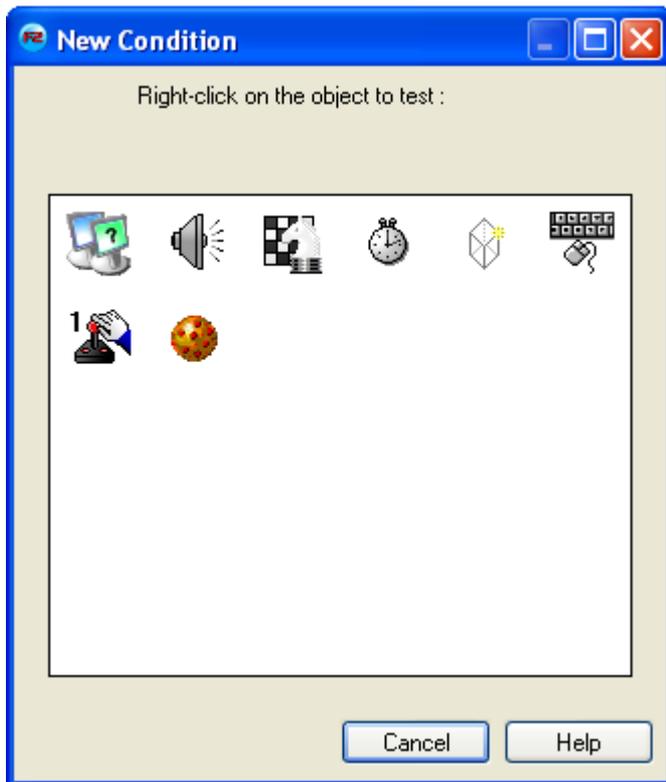


We want the object to act like a bouncing ball. Therefore select Bouncing Ball as the Type. Then give it a decent speed, like 40. No other changes needs to be made now. If you click the Try Movement Button you'll see the ball bounce around in a window. That's just to see how it can look, it wouldn't do that in your game yet. You can see how your game currently runs by pressing F7. You see that the ball just dissapear from the screen.

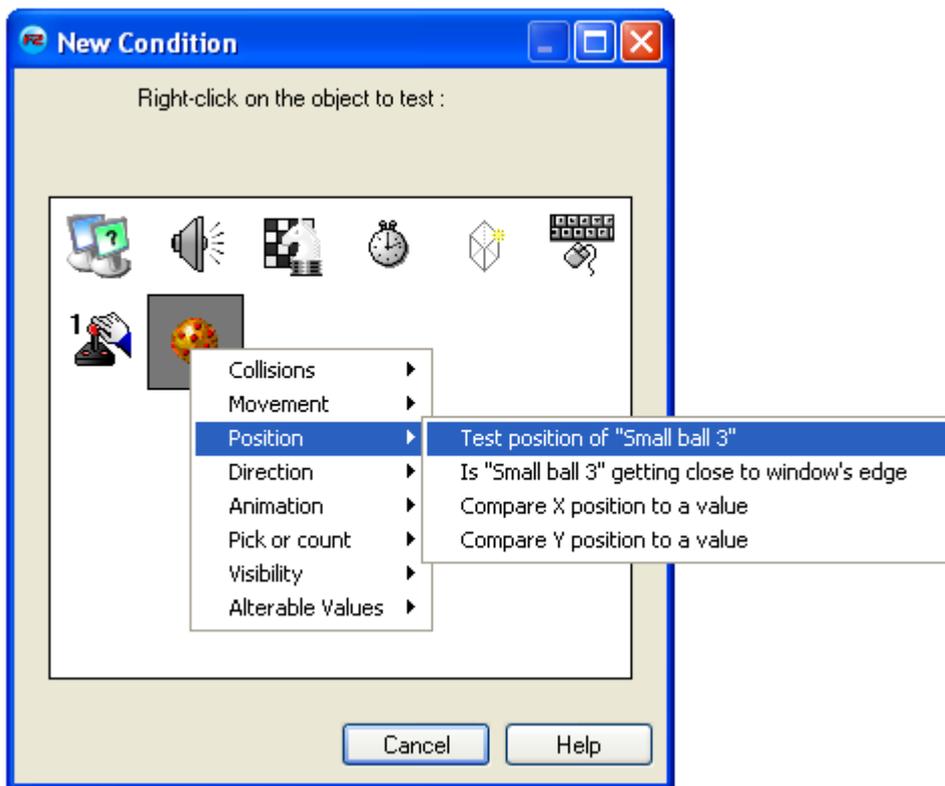
In order to make the ball bounce, we will have to add some events. That's why we will now enter the Event Editor. Select **Event Editor** from the View menu or click on its icon at the top left of the screen. The screen should now look like this:



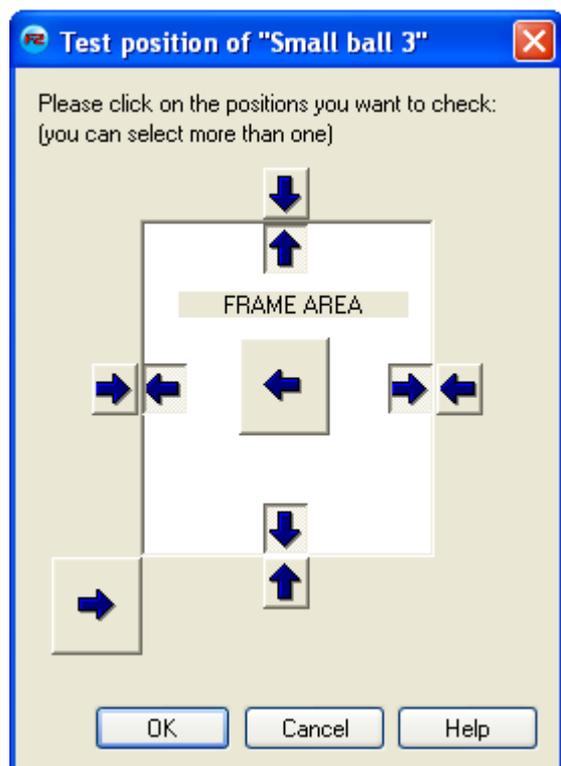
This is where you control all logic in the game. If you want to interact with the objects in your game, this is the place to do that. We want to make our ball bounce, so you need to make a new condition. Click on the New Condition box. You should now see a box like this one:



Right click on the ball and select **Position / Test position of Small ball 3**



You'll see this window. Select all arrows pointing **away from** the center and then click OK.



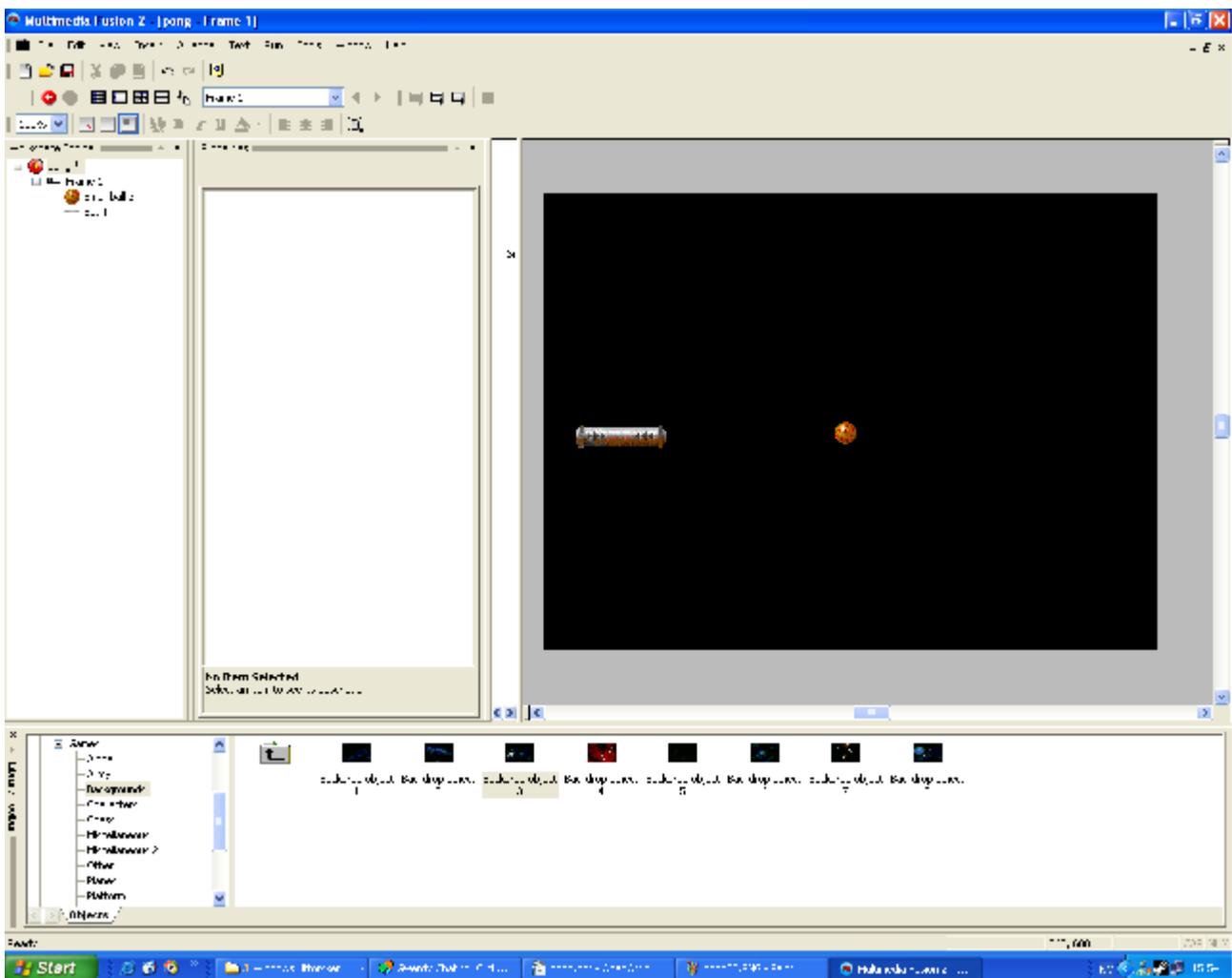
Congratulations, you have made a new Condition! It should look like this:

All the events									
All the objects									
1	<ul style="list-style-type: none"> leaves the play area 								
2	<ul style="list-style-type: none"> New condition 								

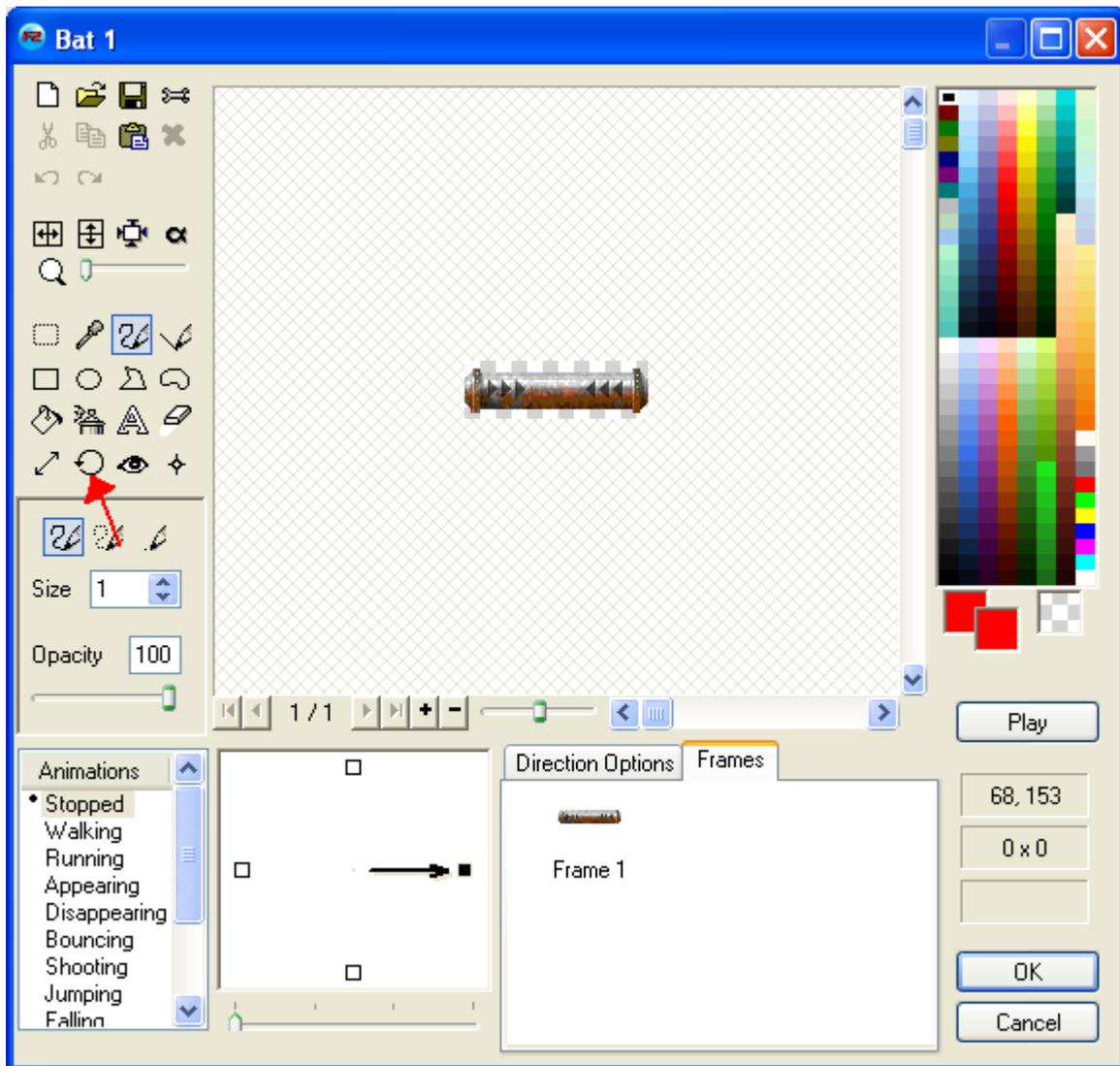
So, now you only have to tell TGF2 what to do when the condition occurs. You see all the white and grey boxes at the right of the condition. Here is where you want to put your actions. Now we only want to do something with the ball. So rightclick in the box below the ball icon. From the list, select Movement / Bounce. And voila! You have made an event with a condition and an action.

Now run the frame by pressing F7 again. The ball should now bounce when it hits the edges.

Ok, now you should know the basics of how to do objecthandling in TGF2. Let's go back to the frame editor! Select Frame Editor from the View menu, or click on its icon. Since this will be a Pong game, we want some paddles for the players to control. So, do like you did when you added the ball. In the same gfx library you should see a nice collection of paddles. Drag one of them into the play area.



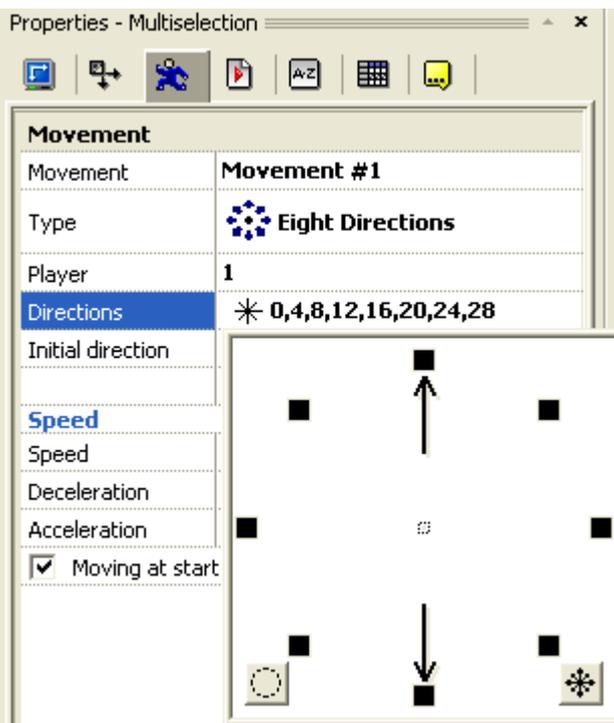
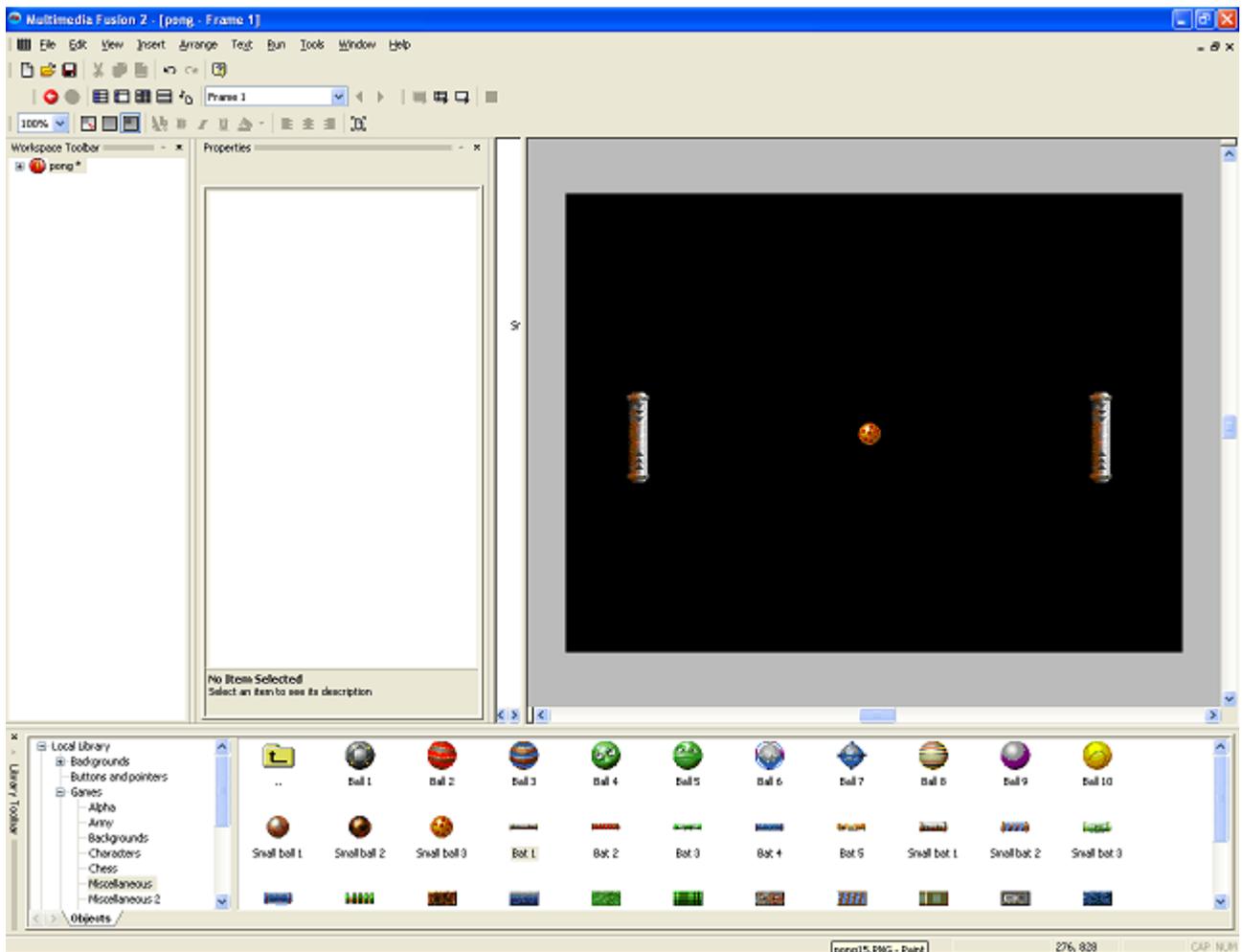
You see that it's angled a little different than we want. Pong should be played left – right, so we need to change the angle of the paddle. There are several ways we can do that. I'll show you one of them. Doubleclick the paddle. You will be presented with the screen below:



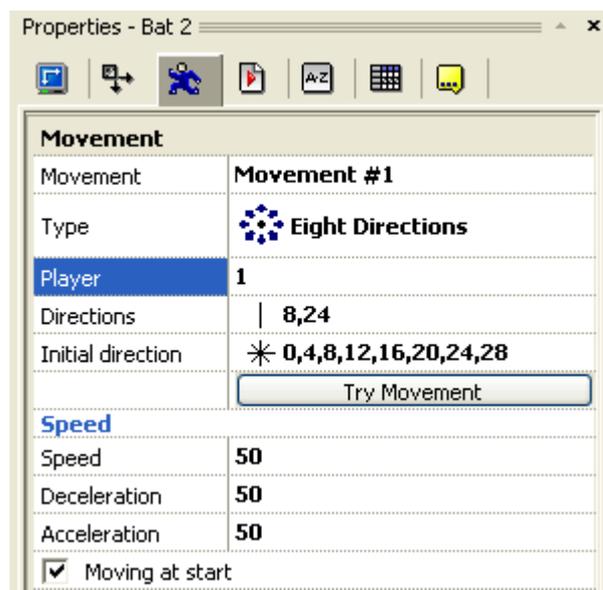
Click on the Rotate button. I drew a little arrow in there to show you where it is. Now you see some new buttons. Click on the 90 degrees button to rotate the object 90 degrees. You'll see that the object has rotated. Click OK to go back to the Frame Editor. Now the paddle should be correct.

We want two paddles. You could drag one more, but we want to **clone** the first one. To clone an object, rightclick on it and click on **Clone Object**. Click OK in the new window.

Now you have two identical objects. But they are two different objects, they just look the same. Place the objects where you want them to be.

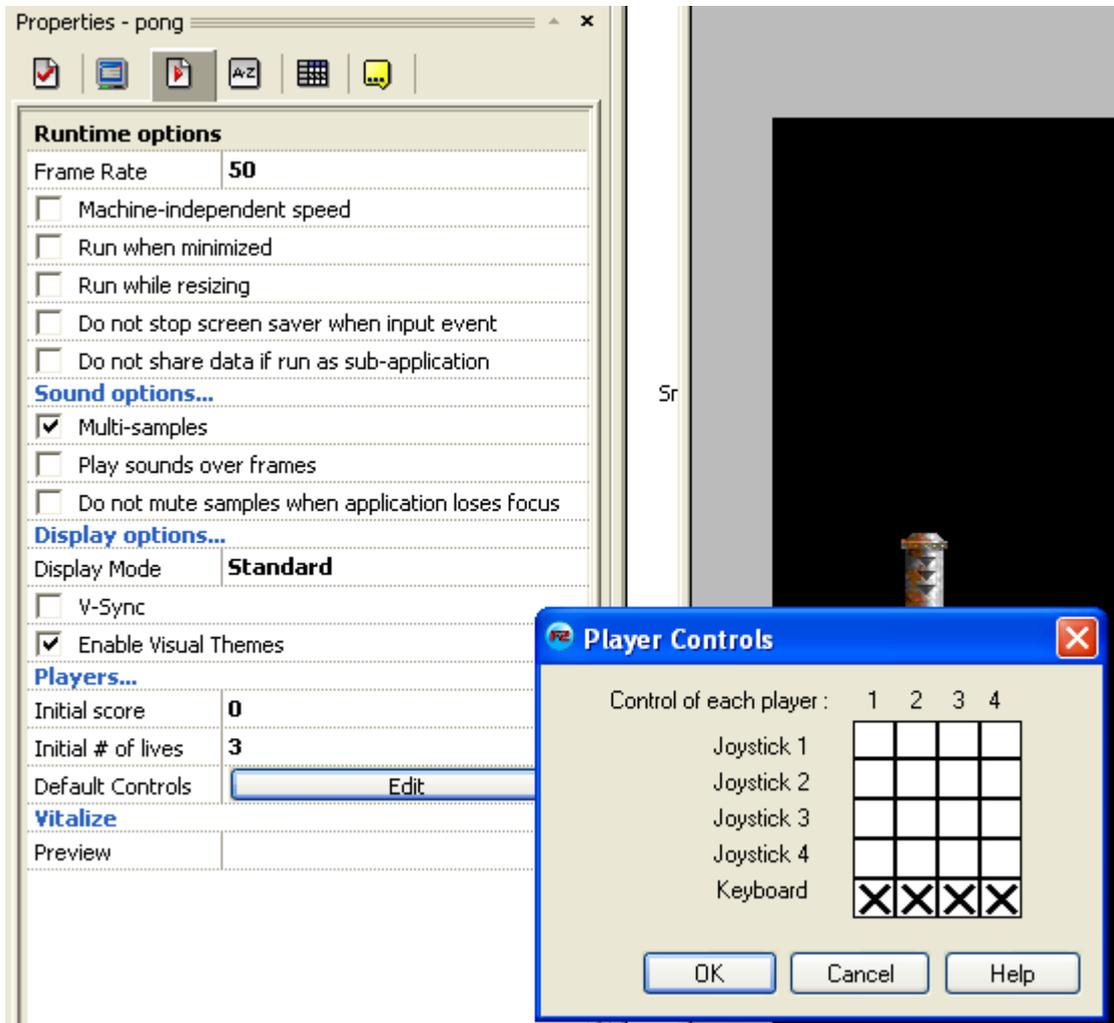


We want to assign movements to these objects so that they can be controlled by the players. Since both paddles should have the same type of movement, it's fastest to select both at once. Click on one of them, then hold **shift** while you click on the other one. In the Movement Properties, set its movement type to Eight Directions. We don't want all the eight directions, so you have to deselect the ones you don't want. We only want the paddles to move up and down.

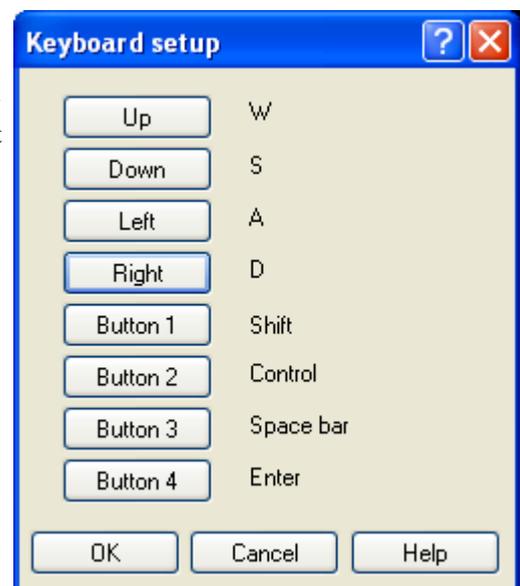


After you've done this you can test the game. Press F7. Do you notice that you can control both players with the arrow keys? This is because both the paddles are set to be controlled by Player 1. To change this go back to the Frame Editor and click on the paddle to the right. In its Movement Properties you'll see that it's set to Player 1. Change this value to 2.

If you test the game again you will still see that you move both players with the arrow keys. This is because you have to define the control for player 2 yourself. In the Workspace toolbar, click on your game's icon. The Workspace toolbar is the one at the far left of the screen. Now select the **Runtime Options Tab**, as shown in the picture below, and click on the big **Edit** button at the right of the **Default Controls** text.

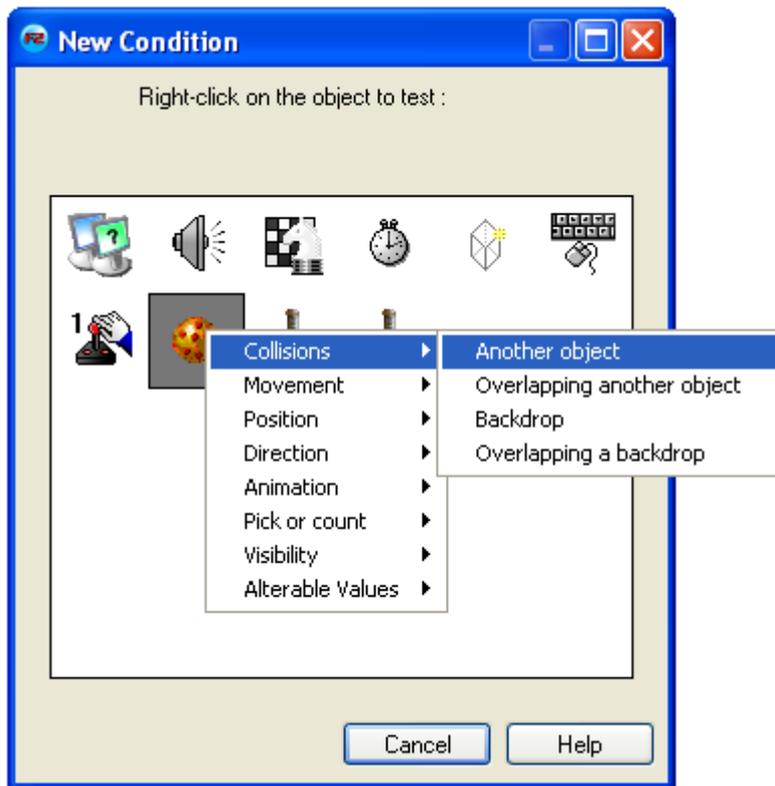


The black crosses tells you that all players are assigned to the keyboard. Click on the cross below Player 2. If you have Joysticks attached to your PC, you can chose that instead. But I chose Keyboard controls. Then just select which buttons to use for Player 2. You only need to select Up and Down buttons for this game.



Ok, test your game again. You can now control each player with different keys. But the game isn't totally working, is it? No, the ball are just bouncing around and not even noticing our paddles. It's time to go back into the Event Editor!

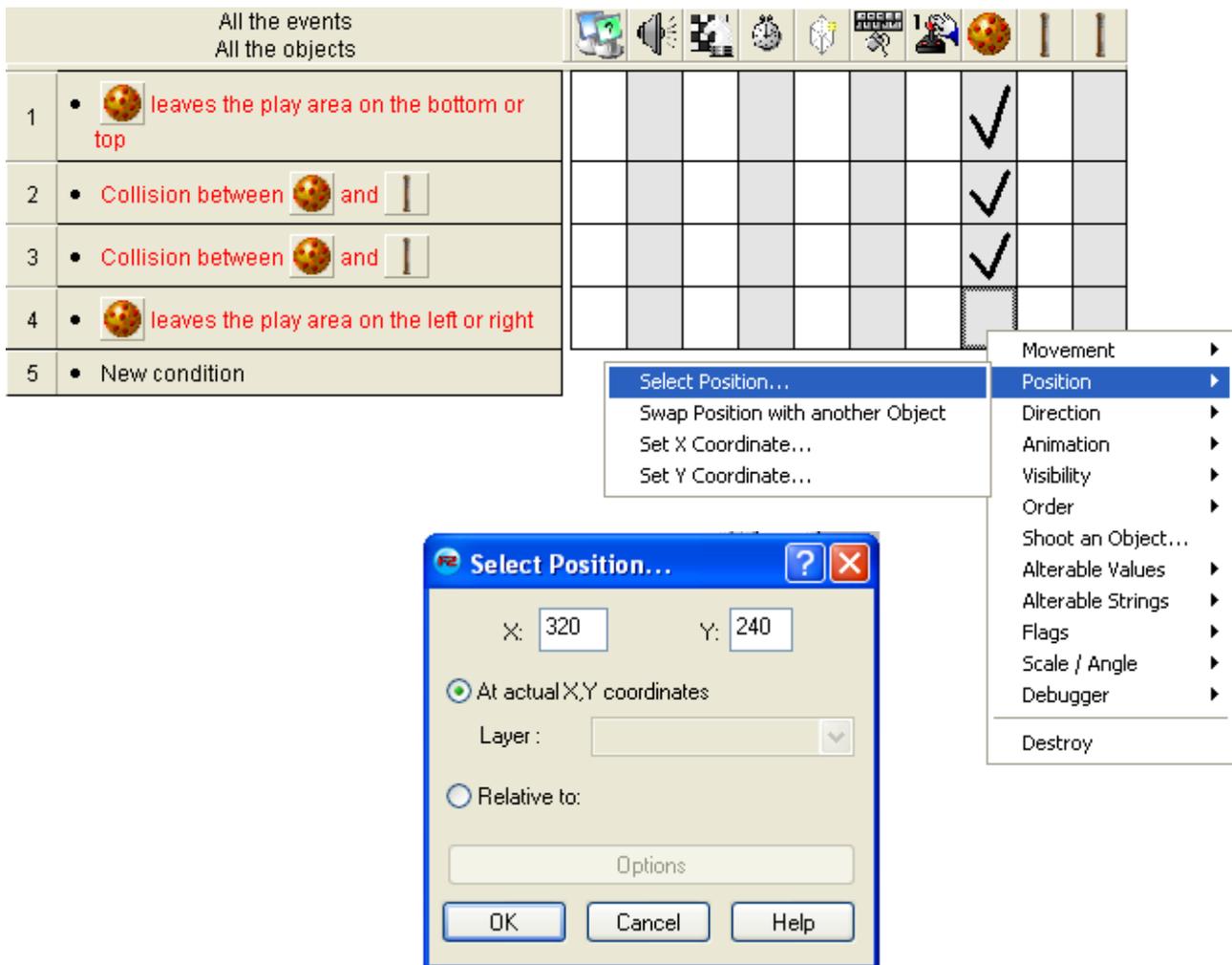
Make a new **condition**. Rightclick the ball and select **Collisions / Another object**.



Select one of the paddles. We want the ball to **bounce**. Since we already have a **bounce action**, simply drag that **action** down to the new **condition**. Or you can add the bounce action manually like the last time.

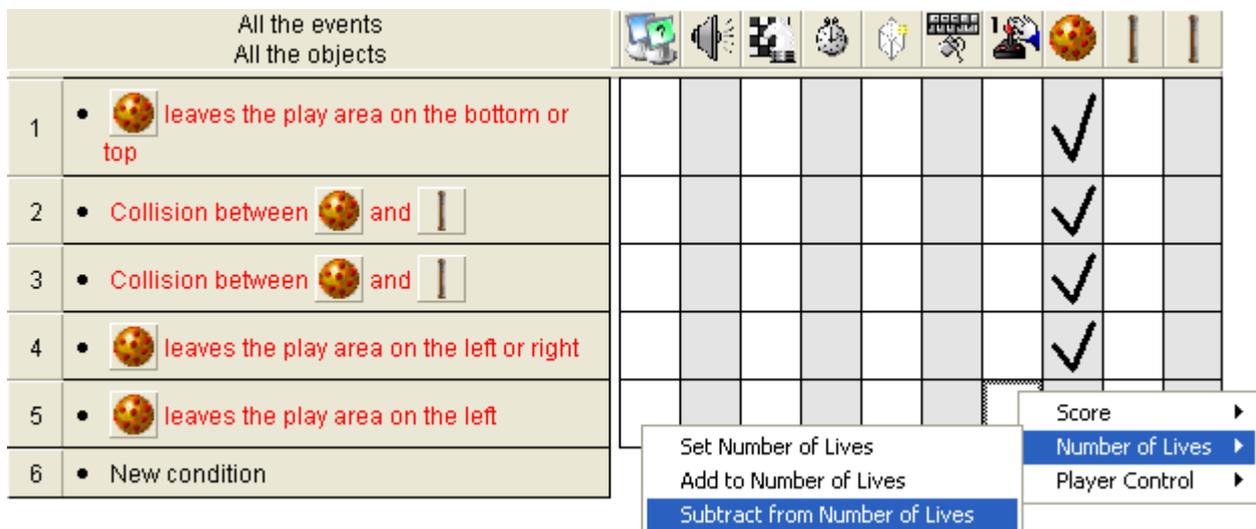
All the events		All the objects									
		Computer	Speaker	Checkerboard	Clock	Lightbulb	Keyboard	Hand	Ball	Paddle 1	Paddle 2
1	• leaves the play area								✓		
2	• Collision between and								✓		
3	• New condition										

Test the game. The ball will now bounce when it touches the paddle. But we want it to bounce when it touches the other paddle too. So do the same thing for that one. A trick is to drag the event you just made down to event 3. Then just doubleclick on the Paddle icon and select the other paddle instead.



The default position should be the center. Check that it says X: 320 and Y: 240 like in the picture. Click OK. Now try the game again. You see that the ball centers itself when one of the players doesn't catch it.

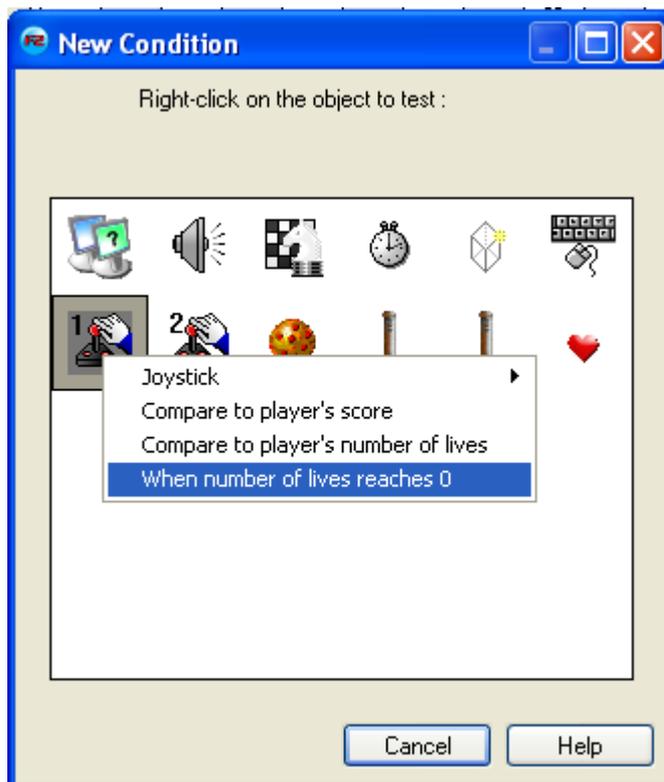
Good! We are almost there now. We only need to add a few more things. We don't want the game do run forever, so we will have to limit the number of times the player can miss the ball. Add a new condition similar to the last one you created. But this time we only want to test for the ball to leave at the left. As the action, select **Number of Lives / Subtract from number of lives** under the **Player 1** icon, and select 1.



Now do the same thing for the other player. But this time test if the object leaves at the **right**.

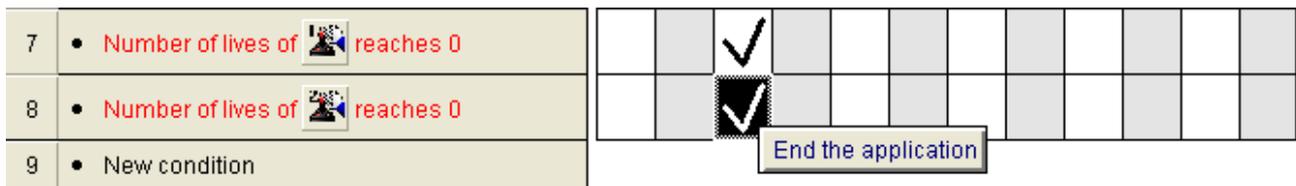
All the events All the objects												
1	• leaves the play area on the bottom or top									✓		
2	• Collision between and									✓		
3	• Collision between and									✓		
4	• leaves the play area on the left or right									✓		
5	• leaves the play area on the left							✓				
6	• leaves the play area on the right								✓			
7	• New condition											

You also have to make an event that tells the game to end when one of the player's lives is 0. So make a new condition that tests for Player 1's lives, like this:



As the action, select **End the Application** from the **Storyboard Controls** icon. Do the same for player 2.

The events should now look like this:



To display the lives in the game, we will have to enter the **Frame Editor** again.

Rightclick somewhere in the **Frame Editor** and select **Insert Object**.

Scroll down and select the Lives Object. Doubleclick on it to create it. Place it above Player 1.

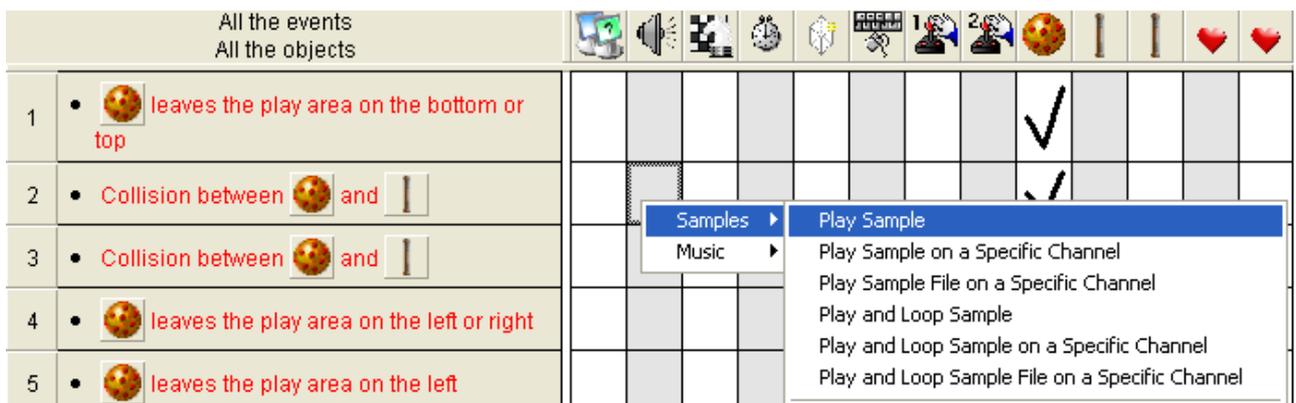
Clone the Lives object. Place it above Player 2 and in it's properties select the **Settings** tab and change it to Player 2.

You see that you can choose how the lives shall appear in the game. Run the game. You should now have a fully functionally 2 player game!

Well, something's still missing? Yes.. Of course we want a background and some sound effects! For the background, just browse through the gfx library again. You'll find some goodlooking backgrounds in Games / Backgrounds / Space – Backdrops. Then just drag the backdrop you want into the playarea. Adding sounds is very easy. Go back to the **Event Editor** again. Now, just add sound actions in the conditions you want to play sounds. For instance this event:



Like this:



You'll be presented with this screen. Click on the **From a file** Browse button. You will find lots of samples to use in the Samples folder in the TGF2 directory.

Congratulations! You have made your first game! I hope you found this tutorial easy to use. If you have any comments, feel free to email me at ilikepopcorn@hotmail.com Also make sure to visit the Clickteam Forum located at www.clickteam.com. Happy Clicking!

Written by Kjetil Nossun (Popcorn)

