CLICKTEAM FUSION

COMPUTING CURRICULUM TOOLKIT

Pong Tutorial

- Beginners' Guide to Fusion 2.5
- Learn the basics of Logic and Loops
- Use Graphics Library to add existing Objects to a game
- Add Scores and Lives to a game
- Use Collisions to allow Objects to interact
- Launch Objects to make a dynamic game experience

Created by: Daniel Block, Clickteam
Adapted from an original tutorial by Pam Dooley
Please see www.clickteam.com for further information about Fusion 2.5

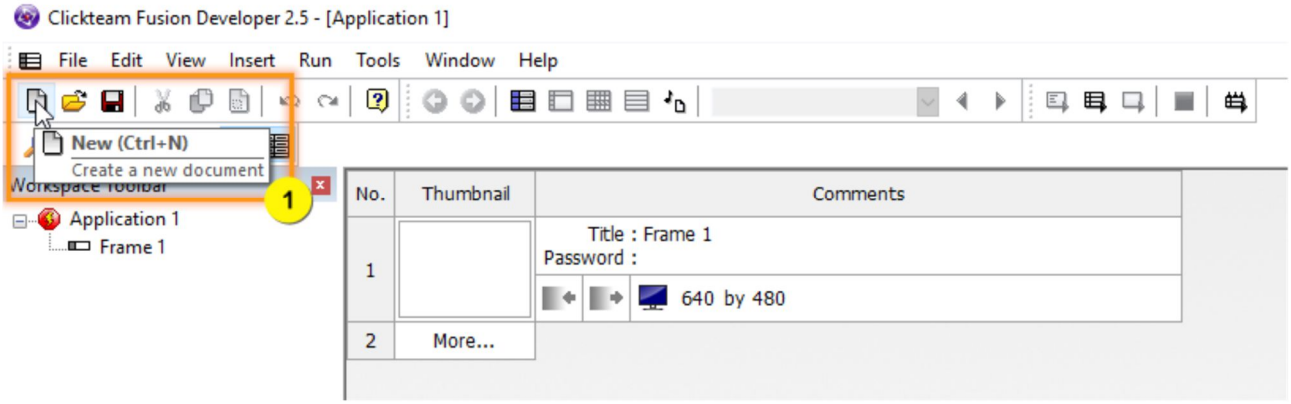
This tutorial will show you how to make your own version of the classic game Pong.

In the first stages, you'll create the Background for your game, the Ball and Paddle Sprites and place them in your game.

Create a New Document

1. To start we need to open Clickteam Fusion 2.5. If the tutorial pops up, close it.

2. On the main toolbar either click **File>New** or just click the **New file icon**



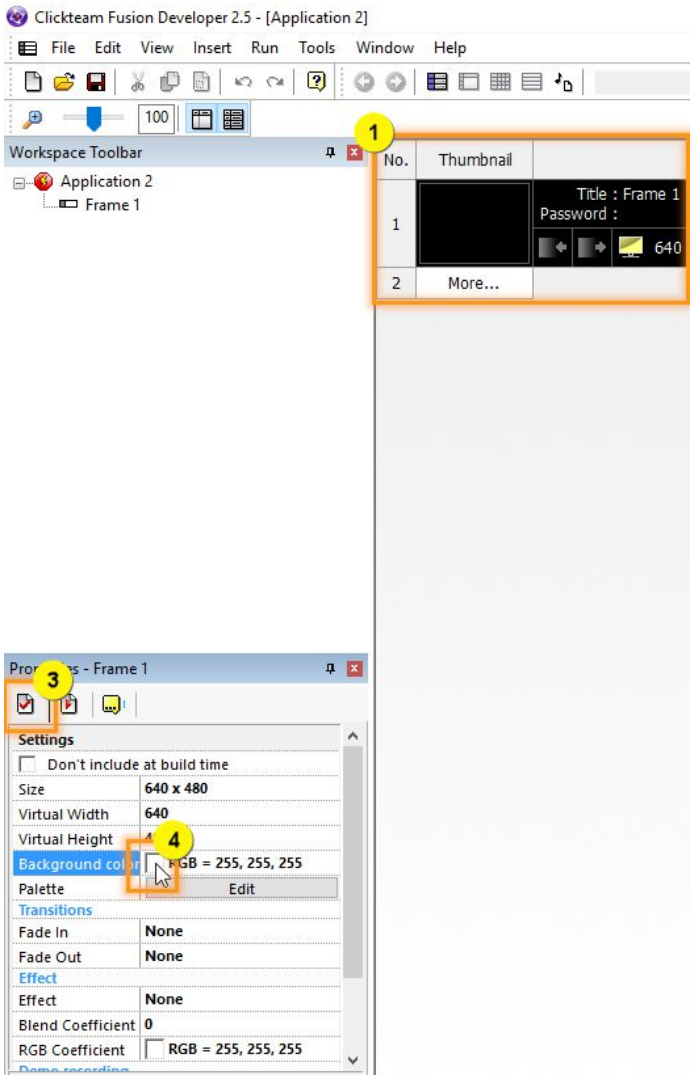
Adjust Frame Properties

1. In the Workspace Toolbar click **Frame 1**.

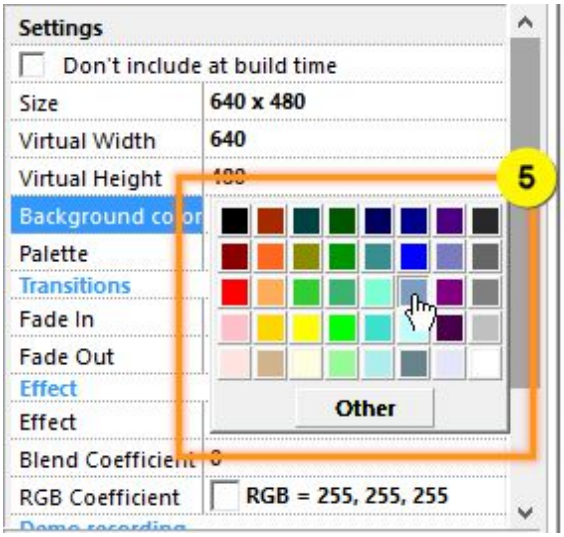
2. You should be able to see the properties displayed in the **Properties** window. If the window is too small, you can drag it larger or move it to the middle of the screen.

3. The **Settings tab** should already be selected but if it's not, click the first button that looks like a box with a **check mark** in it.

4. Click the box between the words **Background color** and **RGB**.

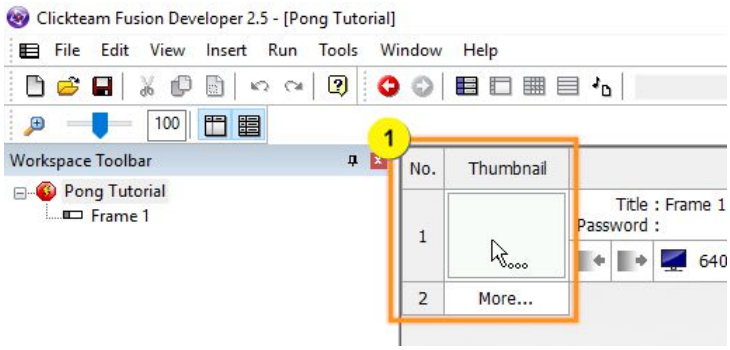


5. Select a colour from the colour palette that pops up by clicking on a square. This will be the colour of the stage. Choose wisely as some colours could be too distracting for a large area.

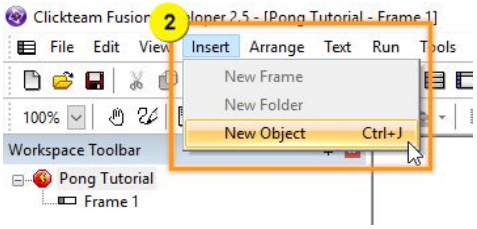


Adding Active Objects

1. Double click the **first frame** to open the Frame Editor. This gives you a big stage of your chosen colour to work on.



2. On the **Insert** menu, click **New Object**.

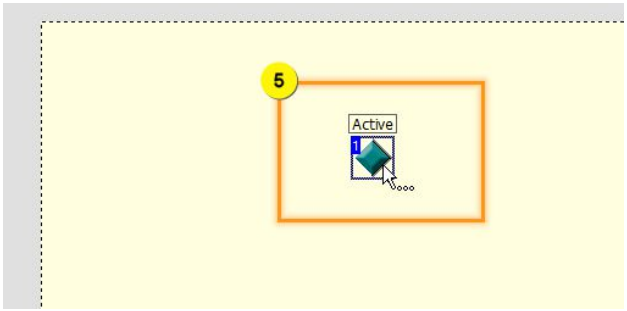




3. In the right pane, click the **Active object** and click **OK**.

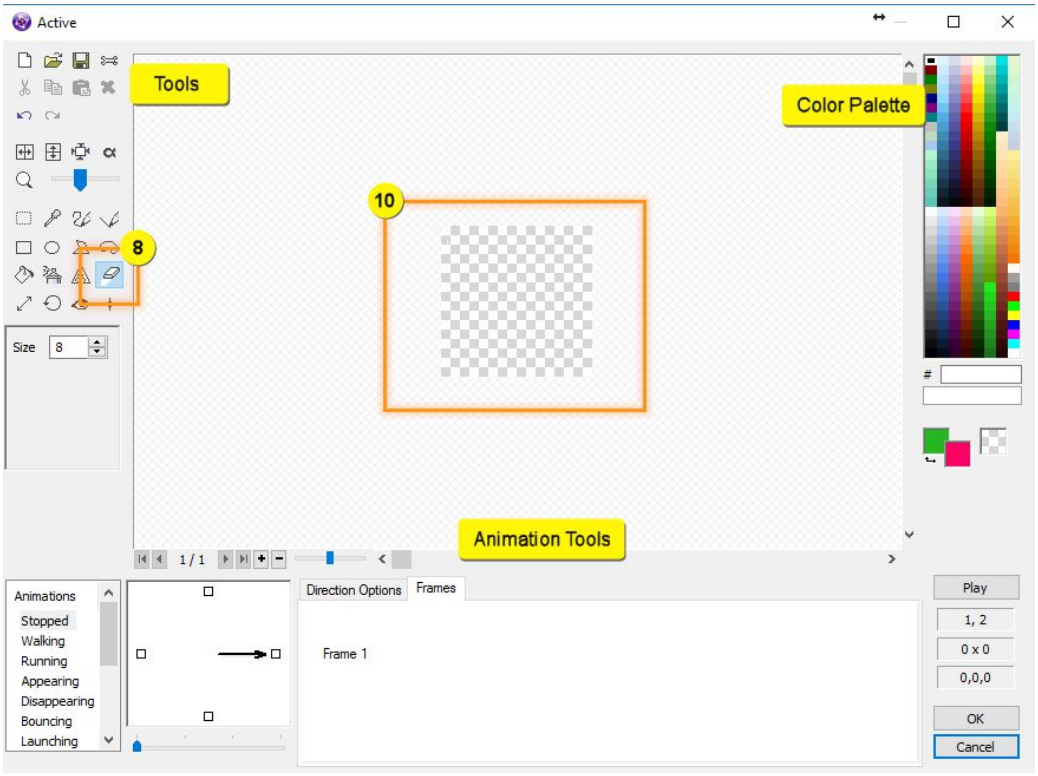
4. Your mouse pointer has changed from an arrow to a cross. That means it is ready to place the Active object. Click anywhere on the coloured stage area.

5. You should see the Active object appear in the frame. It will look like a turquoise diamond. Not quite the look we want for our ball so we will edit it. Double click the **Active object**.



6. This will open the Animation Editor. We are not going to animate this drawing but this is a great place to quickly create simple graphics for the game.

7. The first thing we want to do is get rid of the turquoise diamond sprite.



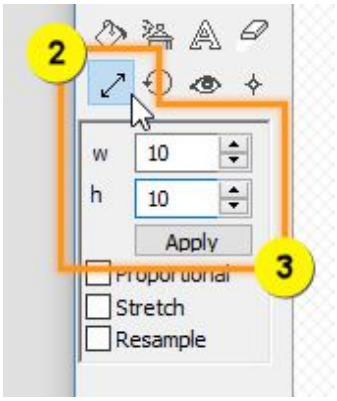
8. On the toolbar on the left, find the **Eraser** tool and click on it.

9. Hold down your mouse button and drag it back and forth over the diamond to erase it.

10. Once you've erased everything, you'll be left with a grey and white chequered square. They indicate the transparent parts of the image. These will not show up in your game.

Create the Ball Graphic

1. If you want a larger ball, you can resize it. Right now the image is 32 x 32 pixels.



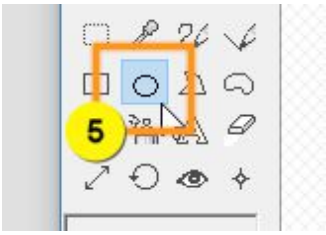
2. On the toolbar, click the **Size** tool.
It is a two-headed arrow.

3. In the box below, change the width to 10 and the height to 10. Click **Apply**.

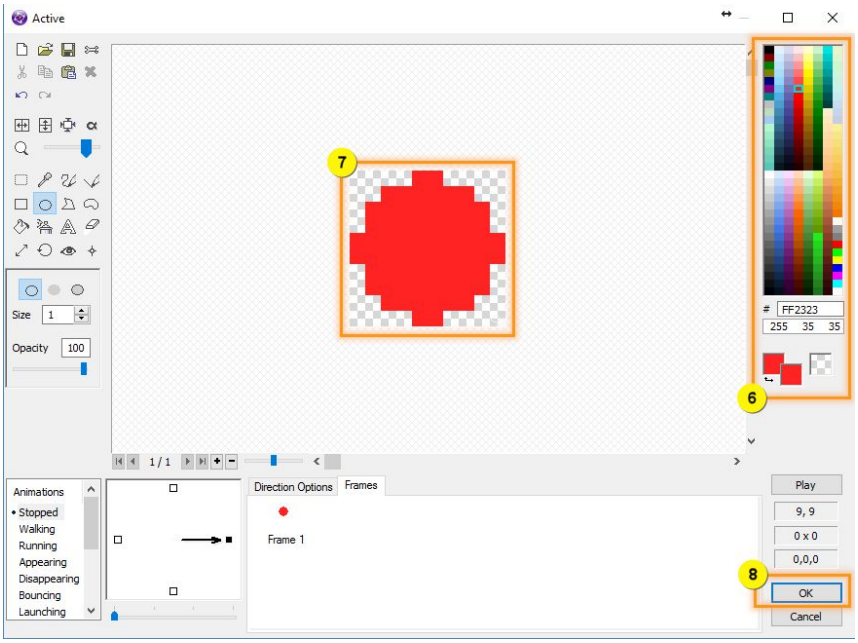
4. Now we'll draw the ball. Is it too small to see? You can zoom in by moving the **zoom lever** to the right.



5. Click on the **ellipse (circle) tool**.



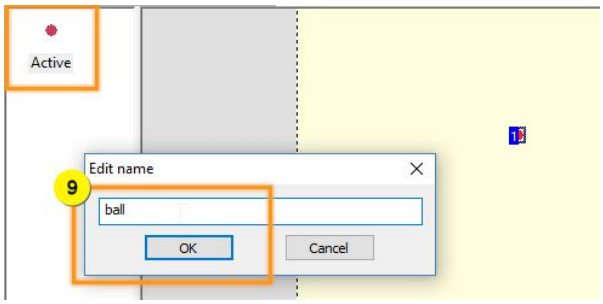
6. In the colour palette on the right, click on the colour you'd like the ball to be. Make sure you choose a colour that stands out on your background. You have to be able to see the ball during game play.



7. Start at the top left corner of the drawing area and drag towards the bottom left corner to draw your ball. Since we are zoomed in, it will look pixellated. All sides of your ball should show in the chequered area.

8. Click **OK** and you should see your little ball on the stage.

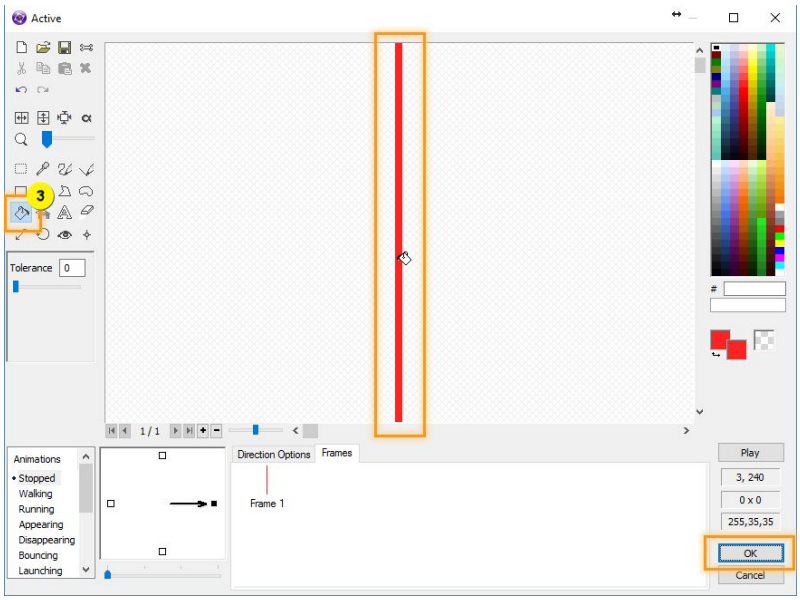
9. Your sprite is named "active". You should change it to be named "ball". Right click on the **"Active" object** in the Object List on the left side of the stage and choose **rename**. Type in the name **"ball"** and click **OK**.



10. **Save** your game as **"Pong"**.

Creating the Net and Paddles

1. Insert another active object (see steps 2-12 in the Adding Active Objects section on page 3).



2. The steps for creating a center line (net) are the same ones you just followed to create the ball, with a couple of changes. **Resize** the object to a **width** of **8** and **height** of **480** (the height of the frame).

3. You do not have to draw a shape because you will just fill the object with the colour you want using the **Color palette** and the **Fill tool**. Click **OK**.

4. **Rename** your active object as **"net"**.

5. Follow the same instructions to create your first paddle but use a **width** of **10** and a **height** of **60**. You can alter these measurements if you want. The smaller the paddle, the harder the game.

6. You'll only need to make one paddle. In the next few steps, we'll clone a second paddle. **Rename** your paddle as **"paddle1"**.

7. **Save** your game.

Cloning, Copying and Duplicating Objects

In Fusion 2.5, there are several different ways to replicate objects. It's important to know the difference between each method. All of these methods can be accessed by right-clicking on an object and selecting the method from the menu.

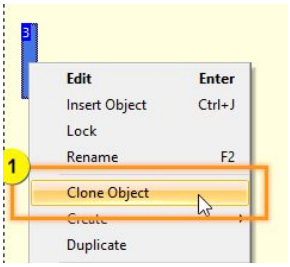
Copy and Paste: When you copy and paste an object, you are creating an exact copy of the original. This copy will have the same settings as the original as well as the same name. If you copy an item and then write code for the item to perform an action, all copies will perform the same action at the same time.

Clone: This method creates a new object that looks like the original but may have completely different settings and has a different name. When you program an item to perform an action, each cloned item must be programmed separately and they will perform their own actions.

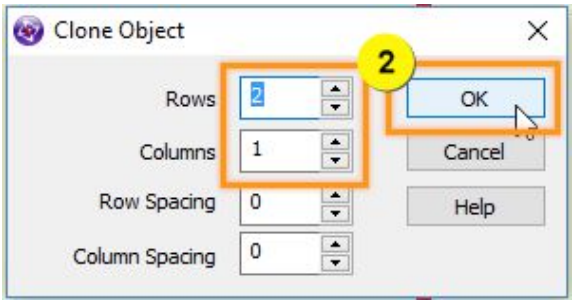
Duplicate: This method is just like copy and paste, except that you can copy and paste multiple objects instead of one object at a time (as you would with copy and paste).

Cloning Objects

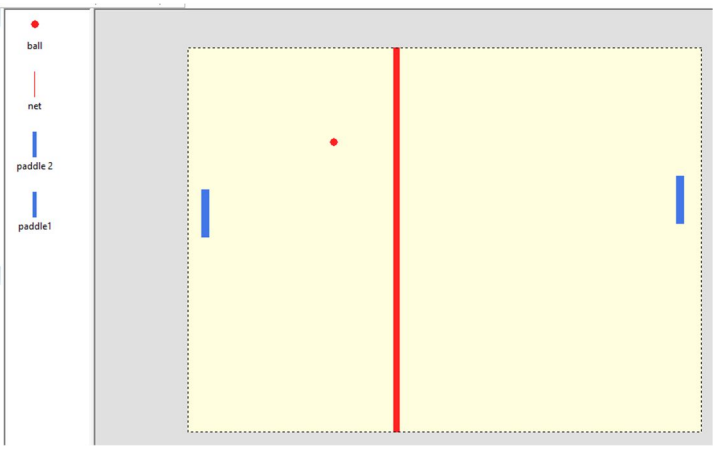
1. Right-click on your paddle and then click Clone object.



2. In the Clone object dialogue box, leave Rows set to 2 and Columns set to 1. Click OK. Now you have two paddles.



Editing Objects



1. Your stage should look similar to this (with your own colours, of course).

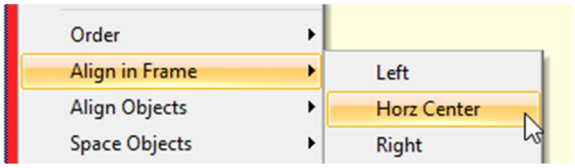
2. If you don't like the colours, size etc... of any of your objects you can double click the object to edit it.

3. You can make objects bigger by using the double headed arrow, changing a size and setting it to "stretch".

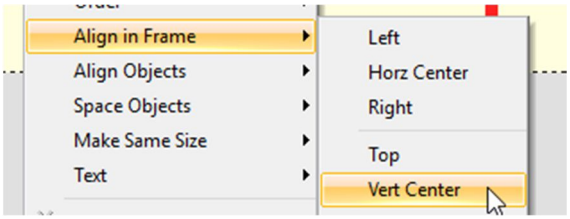
4. You can change colours with the fill bucket.

5. Or you can erase the object and make a new one.

Arranging Objects



1. Right-click the center line. Choose **Align in Frame>Horz Center**. Your center line will now be in the center of the field.



2. Drag the paddles close to the right and left edges of the field but leave a little bit of space between them and the edge.

3. Hold down the **shift** key and select the two paddles. Right click and choose **Align in Frame>Vert Center** so that they are level with each other in the vertical center of the screen. It doesn't matter where you place the ball, as it will be moving all over the screen very soon!

4. You now have your graphics for the Pong game. Save the mfa file. You are ready to program all the events that make the game play.

The Rules

Now you are ready to go back and make your Pong game work. Most of the fun of this game comes from the movement of the ball as it collides with the paddles. Now that we've got the basic elements of the game in position, it's time to start programming the movement of the ball.

This is where you think about your game play. What do you want this game to do? These are called conditions and actions. If a certain condition is met then a certain action takes place. You can also set it up so that if a certain condition is NOT met then a certain action takes place. In programming this is called IF/THEN statements.

1. The ball should start at the middle of the screen and move towards one of the players at a random angle.

2. If the ball hits the top or the bottom of the frame then it needs to bounce back into play heading towards one of the players at an opposite angle (so that it keeps heading towards the person who should hit next.)

3. If a player hits the ball it should bounce back towards the other player, at a random angle.

4. If a player misses the ball then the other person scores a point (you will add that later) and a new ball goes towards the person who missed the last ball from the other player at a random angle.

5. When one player hits a certain score, the game is over.

Moving your Ball

1. Open your **pong.mfa** file.

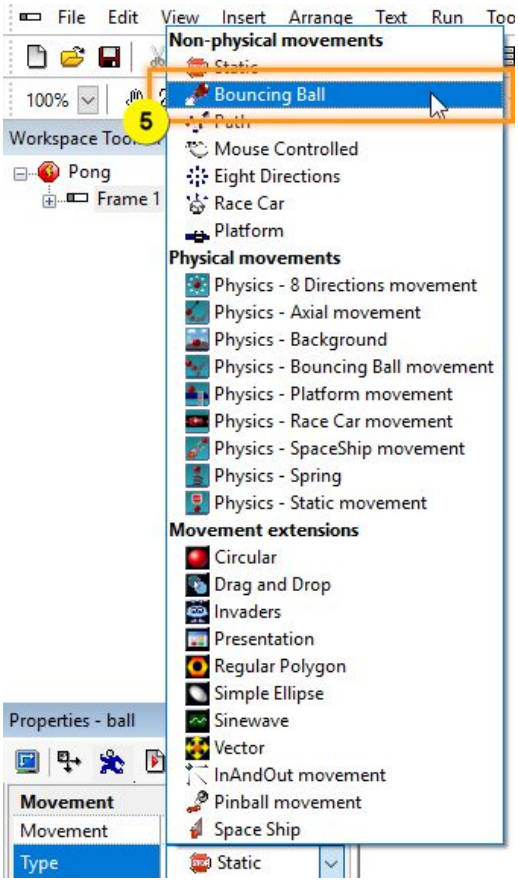
2. Click the **Ball** in the **Frame Editor**.

3. Click on the **properties** panel. Note: If your properties panel disappears, **right click the object** and choose **properties**.

4. In the Properties - Ball window, click the **picture of the running man** in the toolbar.
This is the Movement properties tab.



5. Look at the menu and click the **dropdown menu next to Type**. In the drop-down list, click **Bouncing Ball**.



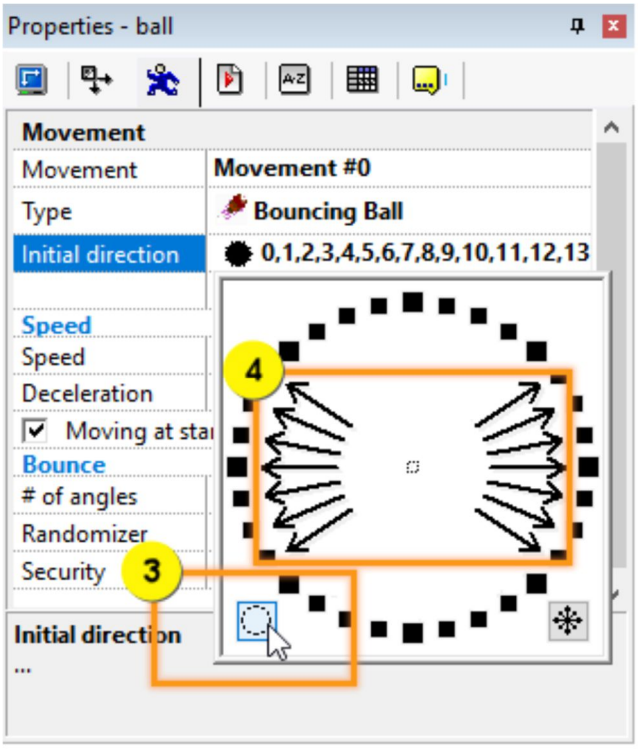
6. The Try Movement button allows you to see what this movement will do once the game is completed. Close the window to stop the preview mode.

Note: This is just a preview; the ball won't actually bounce until we program its actions.

7. There are many options for programming the movement of the bouncing ball.
For example, you will see options for the directions of the bounce, the speed, angles etc...

Controlling Ball Movement

1. In this game it is very important to control the direction the ball bounces. We want the ball to bounce from paddle to paddle.



2. You will start by setting the Initial Direction. The initial direction is the direction the ball goes when the game starts.

3. The easiest way to change these directions is to click on the **list of numbers shown** and then reset these setting so all the arrows disappear. In the lower-left corner of the box, click the small **reset button**.

4. The ball will randomly choose one of the given directions each time the game starts. Set up your initial direction as shown here.

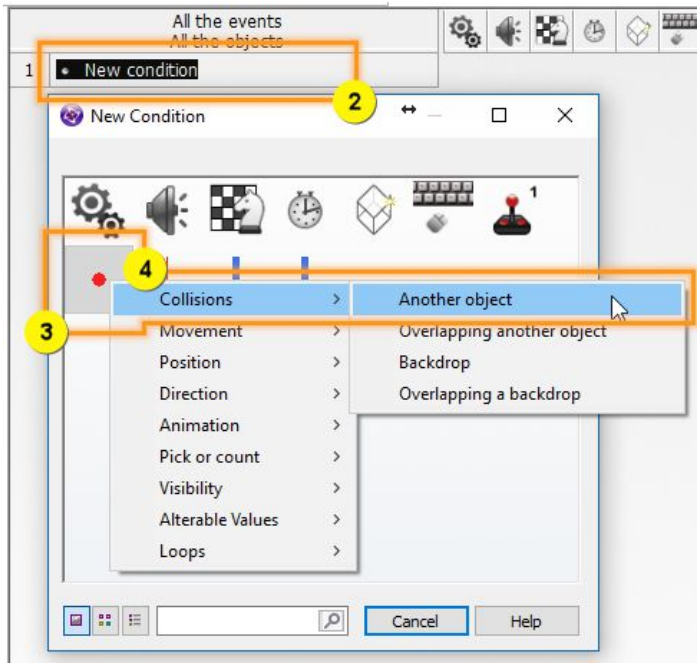
Bouncing the Ball

We will use the Events Editor to program the ball to bounce not only when it hits the paddles but also when it hits the top or the bottom of the frame. That will keep the ball in action longer.



1. Click the **Events Editor** button on the main toolbar.

2. In the Event Editor, click the words **New Condition**.



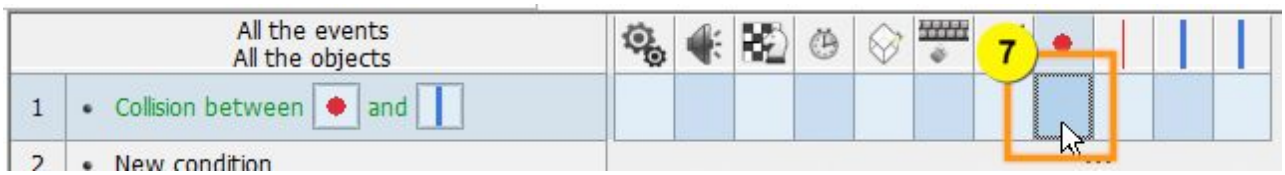
3. Right-click on the **Ball object**.

4. Choose **Collisions>Another Object**.

5. In the Test a collision dialog box, click the **Paddle 1** object and click **OK**.

6. Now we have the condition part of the event taken care of. Let's set the action.

7. Locate the Ball object in the Events Editor.



8. Place your mouse pointer over the **empty box below the Ball object** and then **right-click** your mouse.



9. In the menu, click **Movement** and then click **Bounce**.

10. You should now see a check mark in the box below the ball. When you move your mouse over the check mark, the word Bounce should appear.

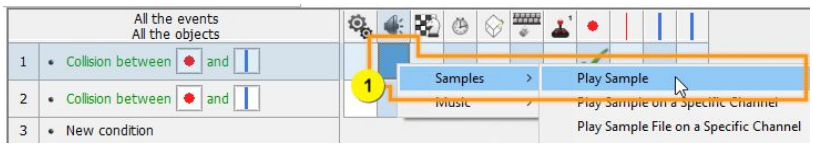
Now make the same type of event for the ball's collision with the other paddle.

Adding Sound Effects

What is a game without sound? When the ball bounces off the paddle you want to hear it hit, not just see it hit. Earlier you made some sound effects to use with this game. Move your audio files into the same folder as you have been saving this game.

Attaching the sound effects is very simple.

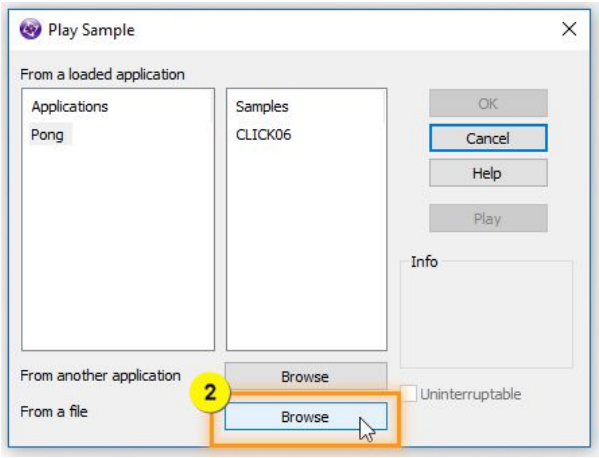
1. Right click under the **speaker icon** for the Collision between the ball and the paddle. Choose **Samples>Play Sample**.



2. Click on the **Browse** button next to "From a file" and navigate to your sound file for that condition.

3. Once you enter a sound file you will see that file listed in your Samples column.

4. You can add your sound effect for the other collision between the ball and paddle by following step 1 and this time just choosing the file from the Samples column.



5. **Save** your game.

The next step is to set up the top and bottom of the frame so that the ball does not leave the stage.

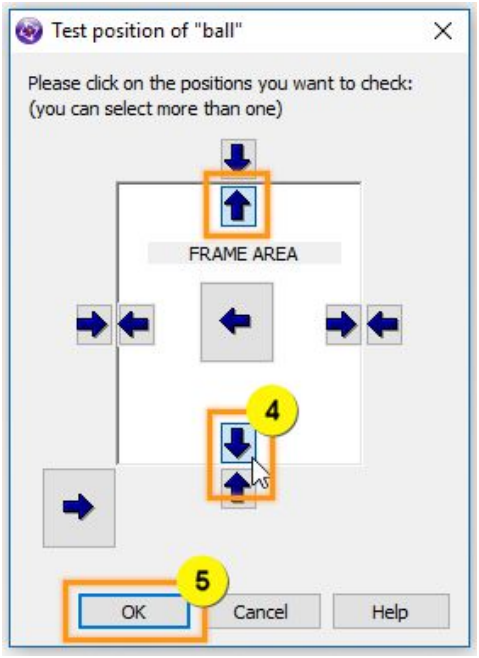
Bouncing the Ball off the Top and Bottom of Frame

1. Open the **Event Editor**.

2. Click **New Condition**.

3. Right-click on the **ball** and then click **Positions**. Click **Test position of "Ball"**.

4. In the Test position dialog box, click on the arrow that is **pointing out of the frame at the top and the bottom**. Make sure it matches the picture shown here.



5. Click **OK**.

6. The condition should state "Ball leaves the play area at the top or bottom." Now
set up an action so that the Ball bounces when that happens.

7. Add your sound effects for the ball bouncing off the top or bottom of the frame.

8. The Event Editor should now look like this. This is a good time to test the game so far.

Test and Save

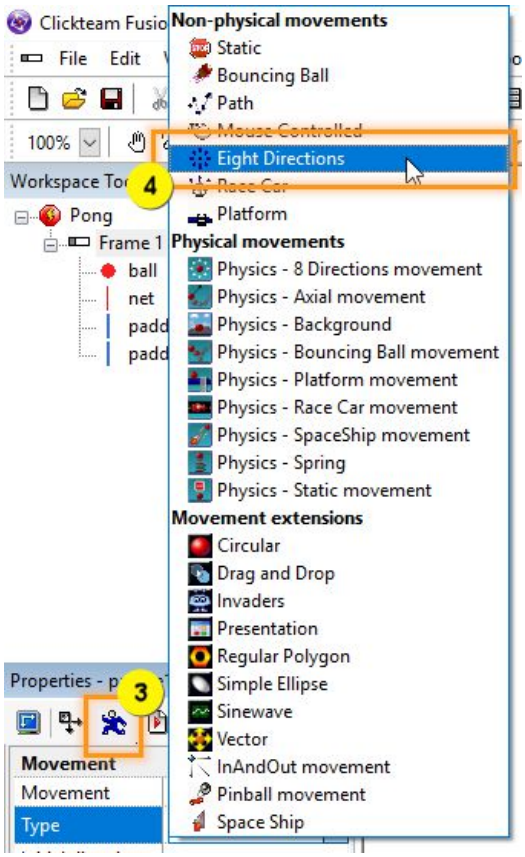
1. To test the game you can either click on **Run>Application** or just hit the **F8** key.

2. The ball should bounce off the top and bottom frames and bounce off the paddles if the ball happens to hit one and should disappear at the right or left side of the screen with a sound effect when you miss the ball. Since you can't move your paddles that should happen pretty quickly. You might have to run the game a few times to see and hear the ball bounce off the top or bottom frames. Just click **File>New** until you see it bounce.

3. This is also a great time to **save** your work.

Moving the Paddles

We've already created the paddles for our players but currently, they won't move. Let's change the movement properties.



1. Go back to the **Frame Editor**.

2. Click on **Paddle 1** in the Frame Editor.

3. Open the properties and choose the **Movement** icon.

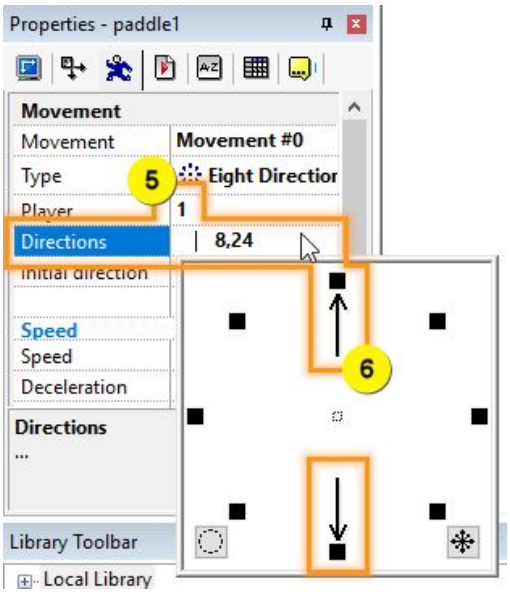
4. Click the drop-down menu for the movement type and select **Eight Directions**.

5. Leave the Player as 1. Click in the line of numbers **next to Directions**.

6. Click the **reset** button and then click the **black boxes for the up and down arrows** only.

7. Set Paddle 2 to have the same movement properties as Paddle 1. You'll need to make one change. You'll need to set Paddle 2 to Player 2.

8. The paddles are now programmed to only move up and down. Later you can change them to also move right and left to make the game more challenging.



The next step is to decide how to move the paddles. Most two person computer games use the keyboard. You need to set up the keys for each person to control one of the paddles.

1. In the Workspace Toolbar, click the name of your game.

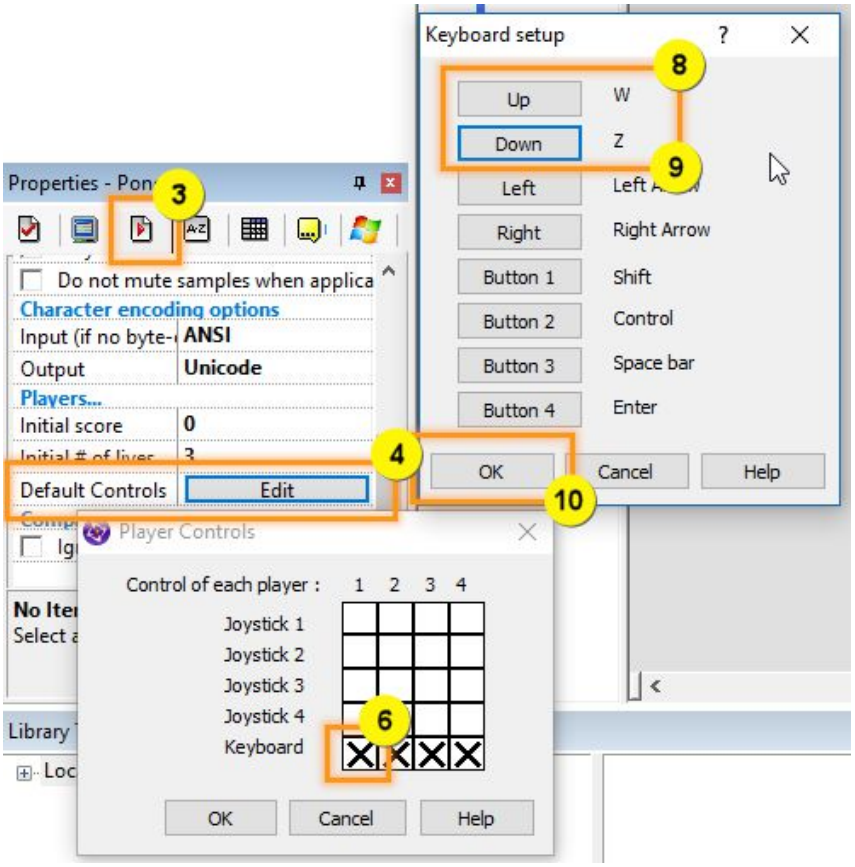
2. Now the properties for the game (not the frame or an object) will be displayed in the Properties window.

3. Click the **Runtime** options tab.

4. Scroll down until you find the heading Players. Click the **Edit** button next to Default Controls.

5. In the Player Controls dialog box, you can set the controls for each player.

6. Click the **X** under **Player 1** and on the same line as Keyboard.



7. On the left is a list of controls. On the right, you can see which key currently corresponds to that control.

8. Click the **Up** button and then press the letter **"W"** on your keyboard.

9. Click the **Down** button and then press the letter **"Z"** on your keyboard.

10. We will leave the rest of the keys alone since our paddles only have two movements. Click **OK** to close Player 1's controls.

11. Click on the **X** for Player 2. Since the default for Player 2 are the up and down arrows, we don't need to make any changes for Player 2. Click OK to close the Player Controls dialog box.

12. Save and test your game.

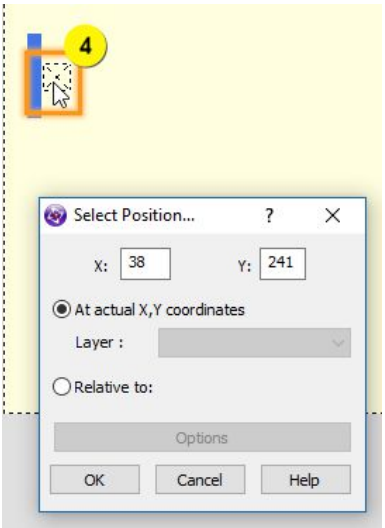
Missing the Ball

First, we'll create a condition. This condition is just like the condition for when the ball leaves the top or bottom of the frame so we won't repeat all the steps here.

1. Create a condition for when the ball leaves the right side of the frame.

2. On the same line as that condition, right-click on the **box below the Ball object**.

3. Click **Position** and then click **Select Position…**



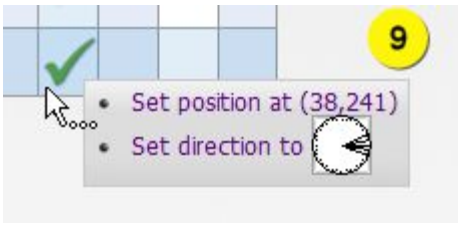
4. You should see a small box with an "x" it in on the background. Drag the box immediately to **the right of the paddle1**.

5. When the ball is missed we want the ball to not only start from the left but also to move towards the right.

6. On the **Ball leaves the play area on the right** condition line, right- click inside the **box below the Ball object**.

7. It already has a check mark for the Set Position action, but we can keep adding actions as needed.

8. Click **Direction** and then click **Select Direction**.



9. In the Select Direction dialog box, choose **directions towards the right side**. Take a look at the picture for an example. Click **OK**.

10. Add your sound effect for when the player misses the ball.

11. Repeat steps 1-10 to create a condition and action for when the ball leaves on the left side.

12. Save and test your game.

Keeping the Paddles on the Stage

When you tested the game, you may have noticed that the two paddles can disappear off the screen. Let's fix that by setting up a condition so that Paddle 1 cannot leave the play area at the top or the bottom. This is just like the condition for when the ball leaves the play area at the top or the bottom.

1. Create a **new condition**.

2. Right-click on the **Paddle1** and then click **Positions**. Click **Test position of "Paddle1"**.

3. In the Test position dialog box, click on the **top and bottom arrows**, just like you did for the ball. Make sure your image matches the picture.

4. On the same line, right-click in the **box under Paddle 1**. Click **Movement** and then click **Stop**.

5. Set up the same condition and action for Paddle 2.



The rest of this project will show you how to add a few more basic game elements, like the score counter and a title screen and the sound effects. No game would be complete without these elements.

Adding Score Counter

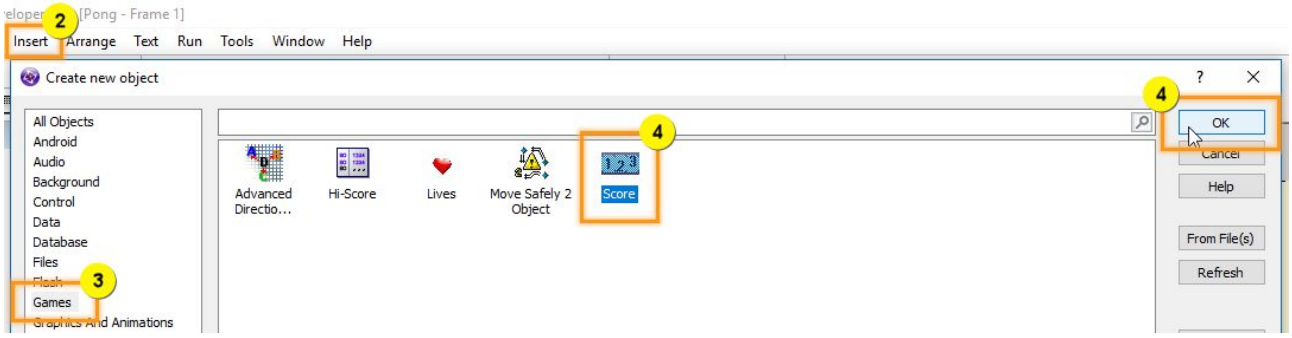
The score counter is special object in MMF2. Unlike other graphical objects you won't find it in the library.

1. Go back to the **Frame Editor**.

2. On the main toolbar, click **Insert** and then click **New Object**. As you can see from the long list on the left, there are lots of objects to choose from!

3. In the list on the left, click **Games**.

4. In the right-pane, click the **Score** object and then click **OK**. Your mouse pointer has changed from an arrow to a cross. That means it is carrying the Score object.



5. Click **near the top of the screen on the left side**. This will be Player 1's score. You should see the Score object in the frame.

6. Rename the score **player1_score**.

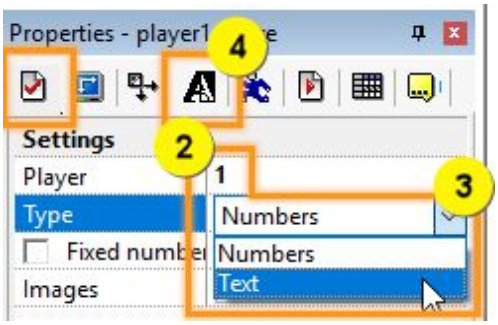
7. Repeat this process and place the other score object on the other side.(Note: Do NOT copy and paste, if you do it will put the same score on each player.)

8. Rename as **player2_score**.

We need to change a few things in the properties window before we can program the rest of the events.

1. Click on the **Player 1 Score counter**.

2. In the **Settings tab**, make sure that Player is set to **1**.



3. Click in the **box next to Type** and change **Numbers to Text** in the drop-down menu. This is so we can easily change the font, size, and color of the numbers.

4. Click the **Text Options tab**. Here you can change the way the number displays.

5. Double-click the box **next to Typeface** to open the Font dialog box. Make sure you choose a font, size and color that show up on your background.

6. Click **OK** when you're done. If your numbers are too big you can resize the score box.

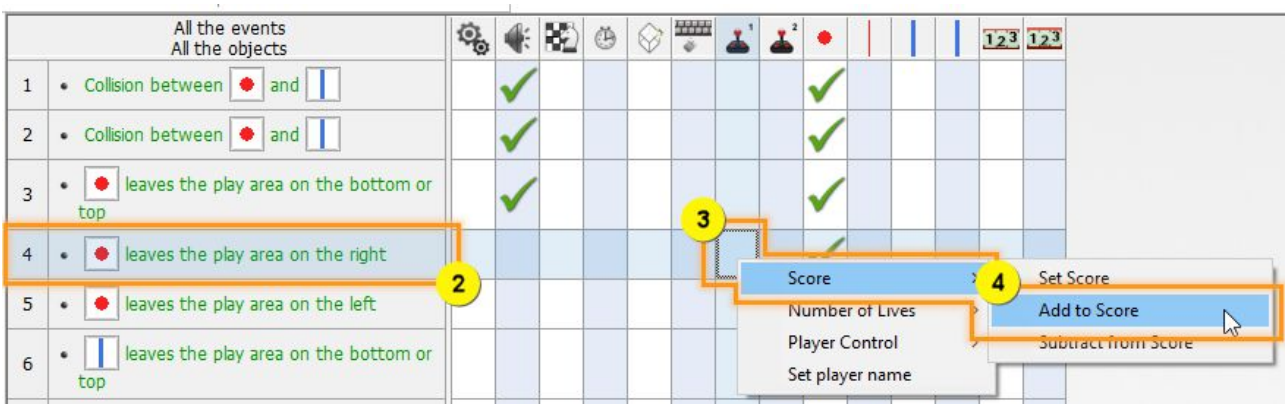
If you test the game now, you'll see that the Score objects don't do anything yet. You need to program them first. You need to add points to the score counter every time the ball exits on the left or the right.

1. Go back to the **Event Editor**.

2. Find the event for when the **ball leaves the play area on the right**.

3. Right-click the **box under the Player 1 object**. It looks like a joystick with a small number one next to it.

4. Right-click in the **empty box**. On the right click menu, click **Score** and then **Add to Score**.



5. In the **Add to Score** window, replace the zero with **1**. Click **OK**.

6. Now repeat these steps for Player 2 when the **ball leaves the playing area on the left**. Be sure to click under the Player 2 object instead of the Player 1 object.

7. Now would be a great time to save your game!

We Have a Winner!

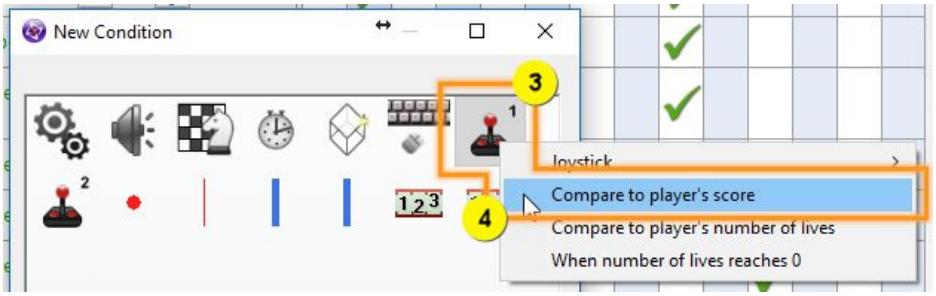
Great! Your game is keeping score! Let's make it so someone can win!

Your game is looking almost finished, but right now no one wins. The score just keeps going up and up. Let's change it so that a player wins when they score 15 points. We'll also display some text that tells the players who won!

1. First, we'll set a condition that checks to see if the score is 15.

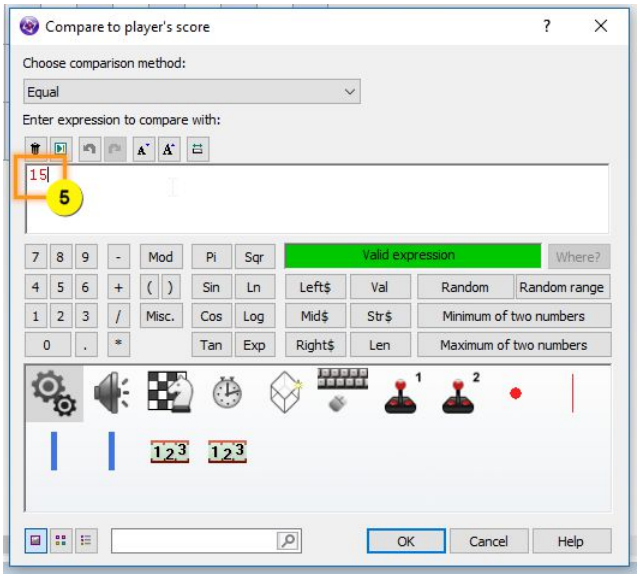
2. Create a **new condition**.

3. In the new condition dialog box, right-click on the **Player 1 object**.



4. Click **Compare to player's score**.

5. Type **15** in the Compare to player's score box, and then click **OK**.



6. Repeat the same steps for Player 2.

7. We have the conditions, if either score equals 15. Now you need the event.

8. The easiest way to end the game is to destroy the ball. When an object is destroyed in MMF2, it is gone and can't be restored until a new game is started.

9. On the Score of Player 1 = 15 line, right-click in the box under the ball object.

10. Click Destroy.

11. Repeat for the Player 2.

12. Save and test your game.

Winner's Screen Page

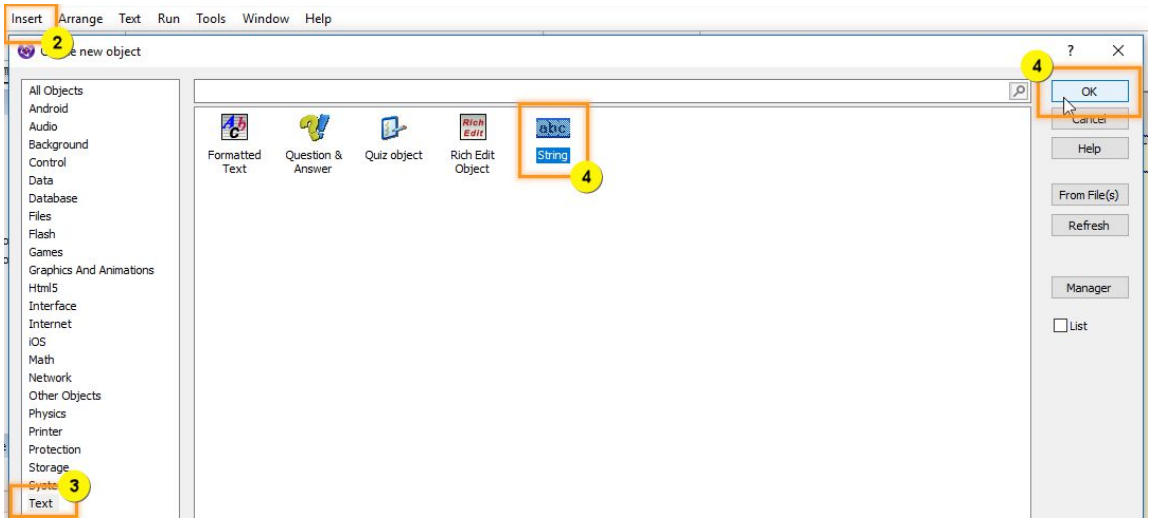
Let's add some text that will display who wins!

1. Go to the **Frame Editor**.

2. Choose **Insert>New Object**.

3. Scroll down and choose **Text**.

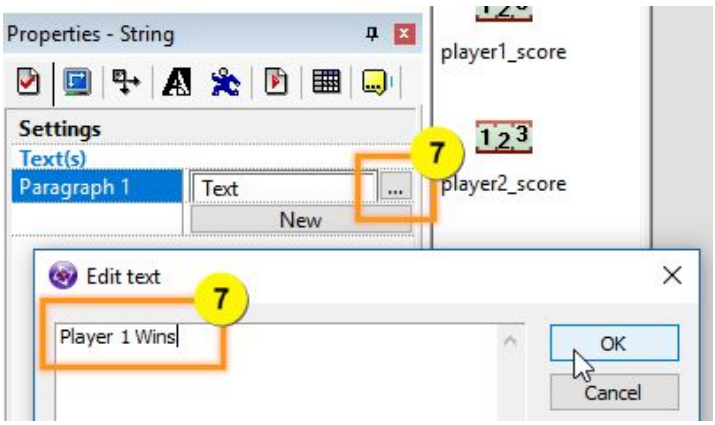
4. In the right-pane, choose **String** and click **OK**. This object lets you display multiple strings of text.



5. Click in the **gray off of the stage** and a text box will appear.

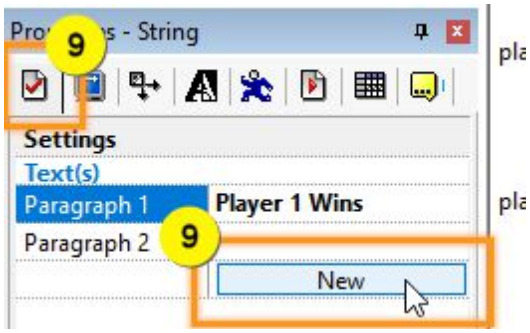
6. Right click the **text box** and choose **Properties**.

7. Click on the **three dots** and a window will pop up where you can type in the text "**Player 1 Wins**".



8. Click on the **Text options button** and format the text to be around **16 px**, change color and font if you want.

9. Click on the **settings** button and click the **New** button.



10. Repeat steps 7 & 8 for the Player 2 Wins paragraph

11. Close the window.

12. Resize the text field so that all the text can be seen on one line.

Programming Winning Text

1. Now that your winning text is set up, let's program it to appear.

2. Go back to the **Event Editor**.

3. On the last line, right-click the **box under the String text object**. It should be the last object in the list.

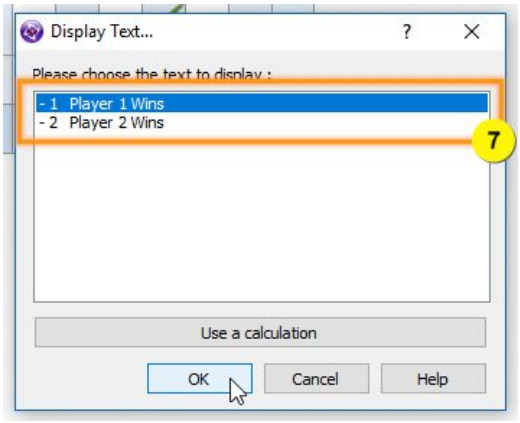


4. Click **Display Text...**

5. You'll need to choose the location of your text. When Player 1 wins, the text should show up on the Player 1 side.

6. Drag the **small box with an X** in the spot you'd like the text to show up. Drag it to the bottom of the Player 1 side.

7. Choose which text paragraph you'd like to display. Since this event is for Player 1 winning, click **Paragraph 1** and then click **OK**.



8. Repeat steps 3-7 for Player 2. Find the line with the condition where Player 2's score equals 13 and then add the action. Remember to select Paragraph 2 when displaying the text.

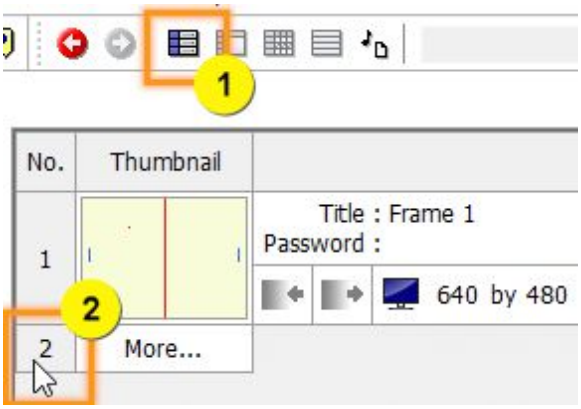
Title Screen

One important aspect of the game is the Title screen and the game directions.

1. Go to the **Storyboard Editor**.

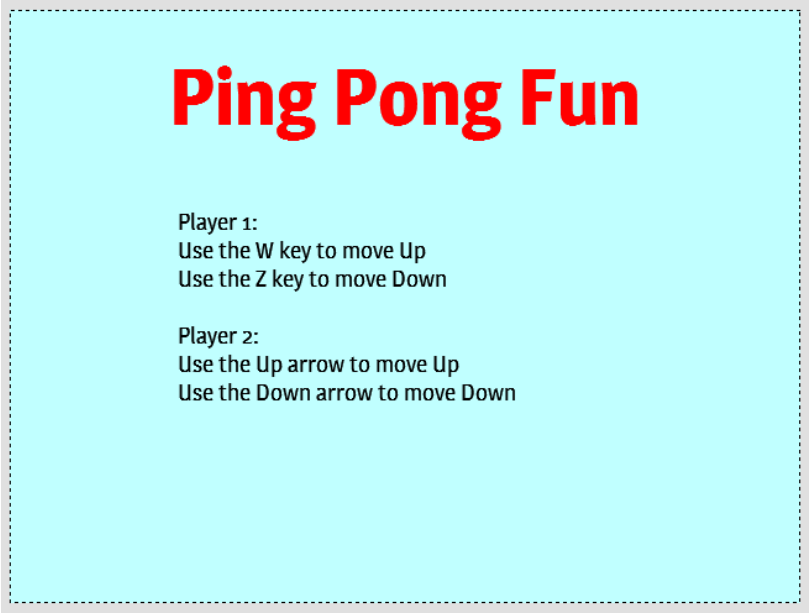
2. Click the **2** to the left of the empty thumbnail.

3. Since this will be our title frame we need to **drag it above Frame 1**.



4. Add your **background and text boxes** (look at previous directions for adding a text box) one for the title and one for the playing directions.

5. Format your text so that it is easily read on your background.



The Start Button

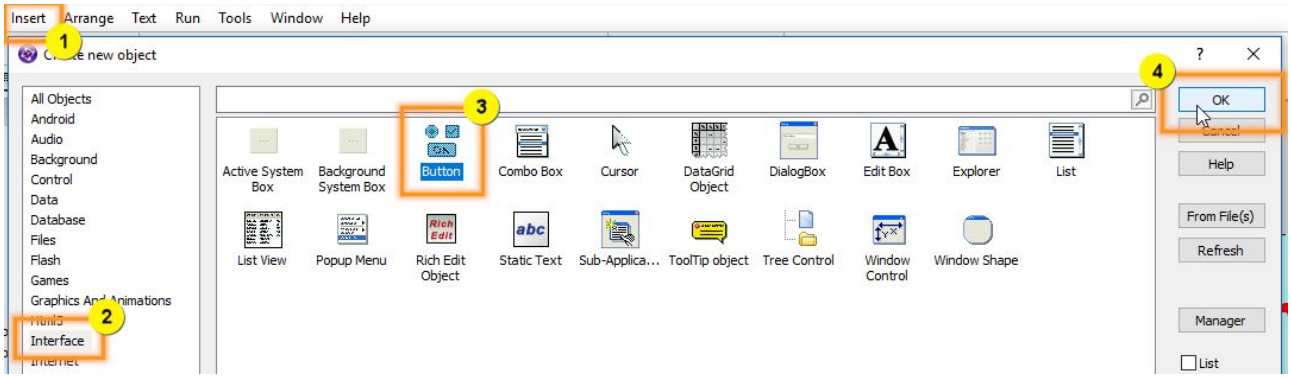
You need a way to move from the title screen to the game.
We will make a start button.

1. On the main toolbar, choose **Insert >New Object**.

2. Choose **Interface** in the list on the left.

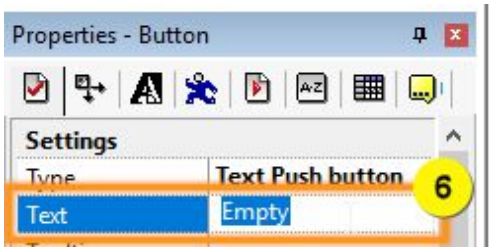
3. Select **Button** from the objects on the right pane.

4. Click **OK**.



5. Click once in the frame to place the button at the **bottom center of the stage**.

6. In the **Properties** window for the button, click the word **Empty** and type **Start**.



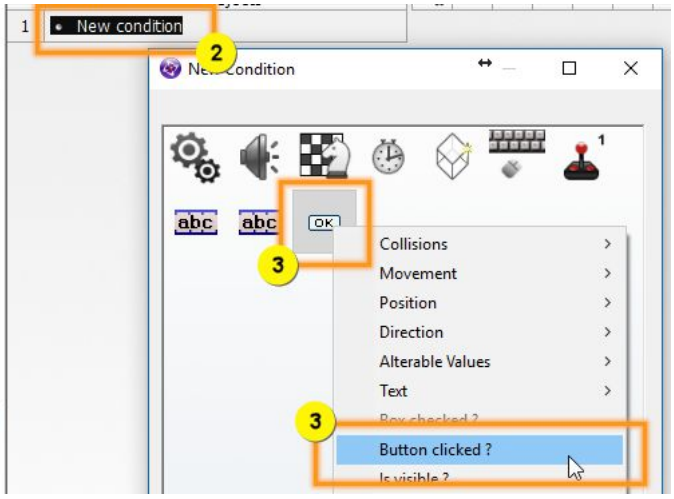
Enabling the Button

1. Go to the **Event Editor** for **Frame 1**.

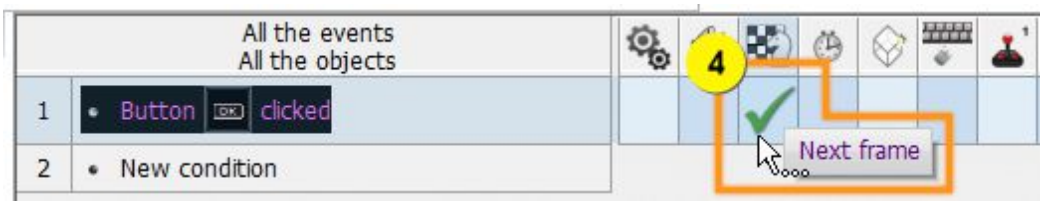
2. Click **New Condition**.

3. Right-click the **Button** object, then click **Button clicked?**

4. In the empty **box below the Storyboard controls** object, right-click and then click **Next Frame**.



5. Save



Your game is now finished and ready to play. Test it out with a friend.

Sharing your Game

In order to share a game with others you have one more step, you need to "build" the game. Building the game saves the game as an exe file and includes all the graphics and sound effects.

1. Choose **File> Build>Application**.

2. Name your game and save. If you make any changes to your game you will need to build the game again.
to need to youwillneed to build the game again.