

INTRODUCTION

In this guide, we will look at how to create your own 'debug' system in Fusion 2.5. Sometimes when you're developing and testing a game, you want to see some of the real-time values of certain strings or numbers, the stuff that's going on behind the scenes. In this guide, I will show you the basics of creating a debug system you can expand upon afterwards to create a more complex one.

Welcome to another guide for Clickteam Fusion 2.5! Some (if not all) of the information in this guide is applicable for Multimedia Fusion 2.0 also. As with all tutorials and guides written in PDF, feel free to display it as a PDF or print to paper. You can study and learn these practices at your own pace, in your own time.

Due to Fusion 2.5's flexible nature of allowing users to accomplish the same result in multiple ways, sometimes there are alternative ways to perform certain routines or functions. These guide(s) will explicitly target the most common, easiest and most effective route.

These hand-written guides are here to help you target or perform a particular routine in Fusion 2.5 enabling you to harness the full potential of what Clickteam Fusion 2.5 can actually do for you as a developer. Whether it's game design or application development you want to do with Fusion 2.5, these guides will hopefully guide you from top to bottom in whatever you're looking to achieve.

This guide has been thoroughly checked before deployment. As always, feedback, comments and suggestions are always welcome on the detail, structure and formatting of these guides. Rather than create individual guides for different learning levels, we hope to be able to reach to all entry level, intermediate and advanced users within the same guide on achieving the same goal.

For more information on how you can download and purchase Clickteam Fusion 2.5, check out the website <http://www.Clickteam.com>

For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.



CONTENTS

1. Introduction to a Debug System
2. Creating the Test Playarea
3. Creating the Debug Layer
4. Implementing the Events
5. Testing the Debug System
6. Final Notes

For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.



1. INTRODUCTION TO A DEBUG SYSTEM

A debug system is a kind of system which you can develop, yourself to see certain values or strings that are not usually accessible when the game is running. Fusion 2.5 has a basic debugger built-in already but this can be limited with certain extensions and doesn't provide much room to see what's going on.

A debug system is not always necessary for all types of games and applications, however if you have a feeling your game engine or application may turn into a complex one, it is handy to have a debug layer so you can see what's going on every single frame. You can grab data in real-time from almost every single object.

In this guide, I will teach you the rudimentary basics to creating such a debug system. It is then up to you to expand and customise this for your future projects. Once you have a grasp on how to create a basic debug system, creating your own customised one will be pretty easy and very flexible.

For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

2. CREATING THE TEST PLAYAREA

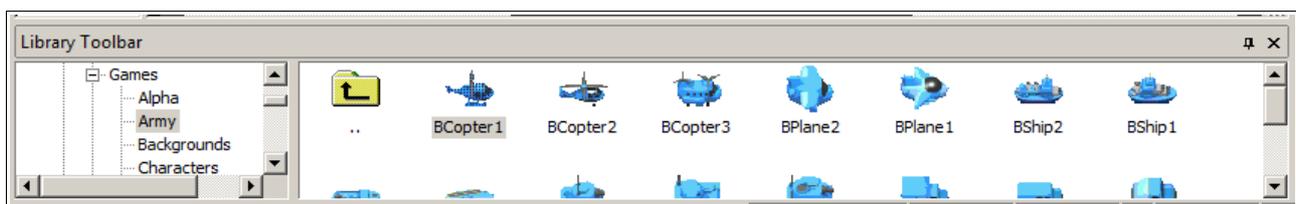
Fire up Clickteam Fusion 2.5. Create a new application. Let's keep it at all the default values (640x480 resolution etc.).

Now, in Frame 1, let's put some objects onto the playarea. Select **View** from the menu, select **Toolbars > Library Window**. This should popup the Library Window in Fusion, the library window contains some pre-made objects and graphics included royalty-free with Fusion 2.5.



Now, repeat this process but in select **Toolbars > Layers Toolbar**. We will need the layers toolbar as we're going to create our initial game playarea on layer 1 and our debug system on Layer 2. The reason behind this is, we want our Debug Layer on TOP of the playarea, so we can see the values we need to.

Now, for this guide, I am going to use the **Games > Army** graphic set:



Let's just drag a few of these into the playarea. It really doesn't matter which ones, just select a few different ones, like a Helicopter, a Plane and a Tank.

Scatter them about in the playarea. Now, we need to assign some movements to these objects. We will take control of the Tank and we will assign some basic AI to the Helicopter and the Plane. Select both the Plane and the Helicopter and give them the Bouncing Ball Movement with the following settings...

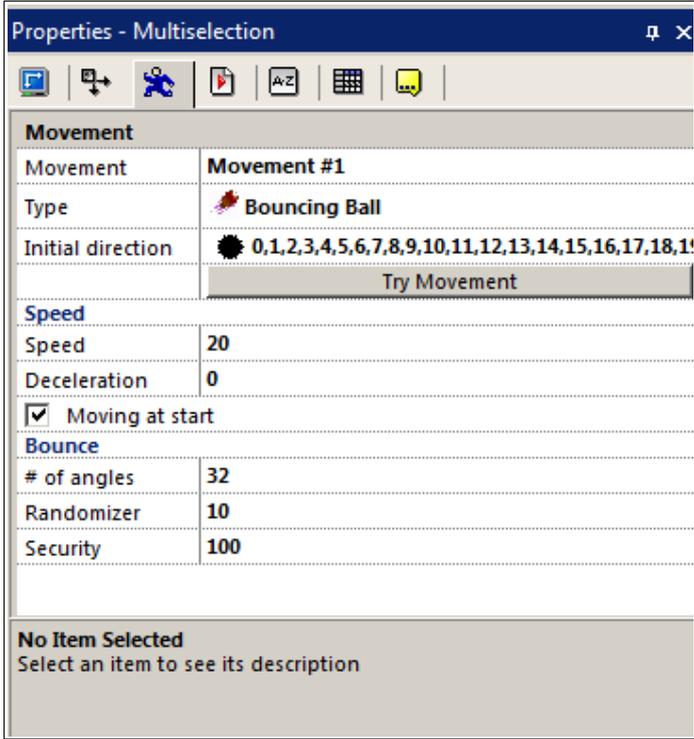
For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

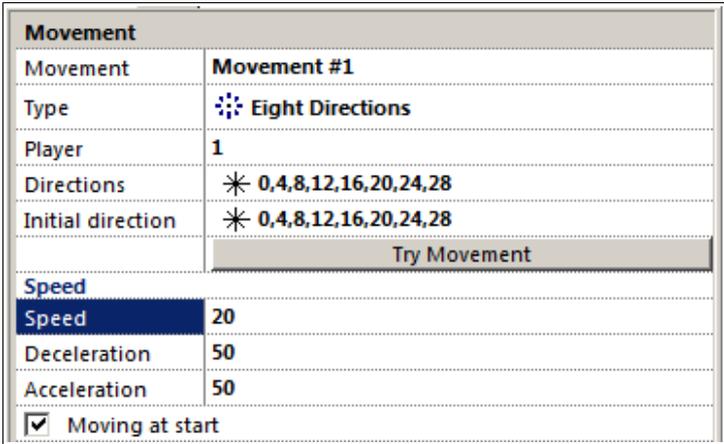
The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

2. CREATING THE TEST PLAYAREA

Like this:



Now, select the tank and give the tank and 8-directional movement:



This means we can now (automatically) move our tank with the arrow keys.

For more information, tutorials, examples and walk-thrus visit the website.

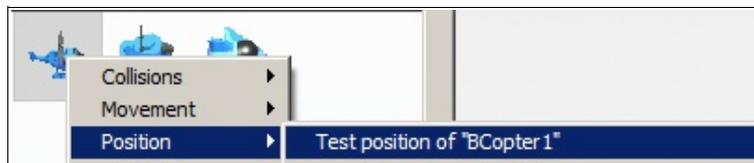
Website: <http://www.Clickteam.com> – Community: <http://community.clickteam.com>

2. CREATING THE TEST PLAYAREA

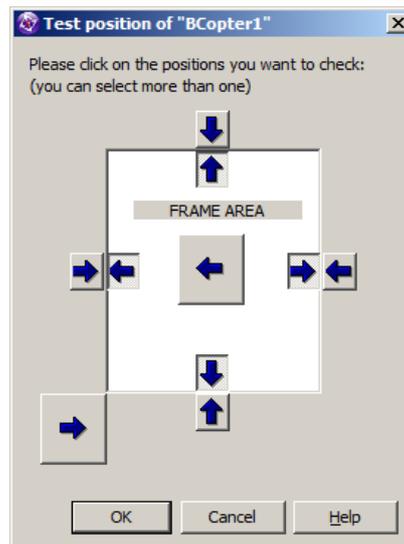
Go ahead and hit F8 on your keyboard to sample this.

Naturally, the objects with bouncing ball movement (Copter/Plane) will exit the playarea. We can stop that with an event, we'll cover now, then it's done.

Jump into the event editor, create a new condition. Hover over the Helicopter, right-click and select, **Position > Test position of "Bcopter1"...**



Once you've done this then select all the possible exit points of the playarea:



Click OK. On that event line, scroll across to the Bcopter1 icon, right-click and select **Movement > Bounce**.



For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.



2. CREATING THE TEST PLAYAREA

Repeat this process for the Plane too.

Once you have done that, Hit F8 again and now you will see the plane and copter will stay within the playarea at all times.

Great, we now have a little test playarea setup. Now it's time to start developing our debug system.

In this debug system, we will detect (in real-time) all these objects, their movements, animation frame numbers and basically anything else we feel like tracking, in real-time.

Let's proceed to the next stage of setting up the Debug System in the frame editor.

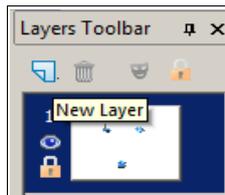
For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

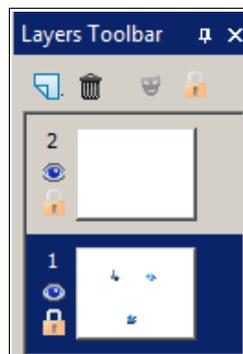
The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

3. CREATING THE DEBUG LAYER

Remember the Layer Toolbar we displayed earlier? It should still be on your screen, to the right hand side. Let's create a new layer, click the New Layer button...



It should now look like this:



Layer 2 is where we will put our Debug System on. Now, Fusion will insert objects into the playarea in the Layer we have selected. As you can see above, we currently still have Layer 1 selected but we want to put our Debug System on Layer 2 so simply Left-Click on Layer 2. It should now show as selected. Click on the EYE icon on Layer 1 to HIDE Layer 1. We don't need to see what's on Layer 1 whilst constructing our Debug Layer on Layer 2.

Now, let's start to implement our Debug System.

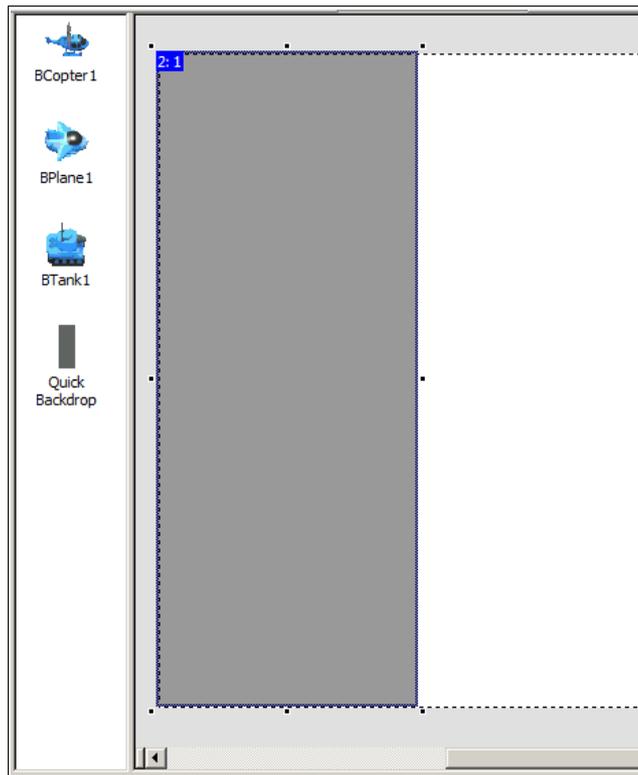
For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

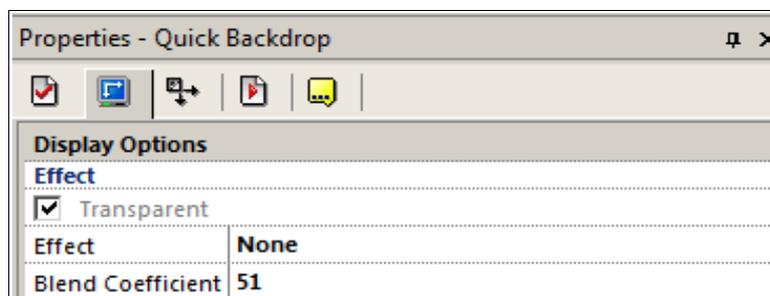
The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

3. CREATING THE DEBUG LAYER

On layer 2, insert a new object > Quick Backdrop. Resize and position it like so:



Select it and hover over to the properties tab on the left hand side. Select the Display Tab and set the Blend Coefficient to 51. This will give it a slight fade (so when we display this layer, we can still see what's going on underneath).



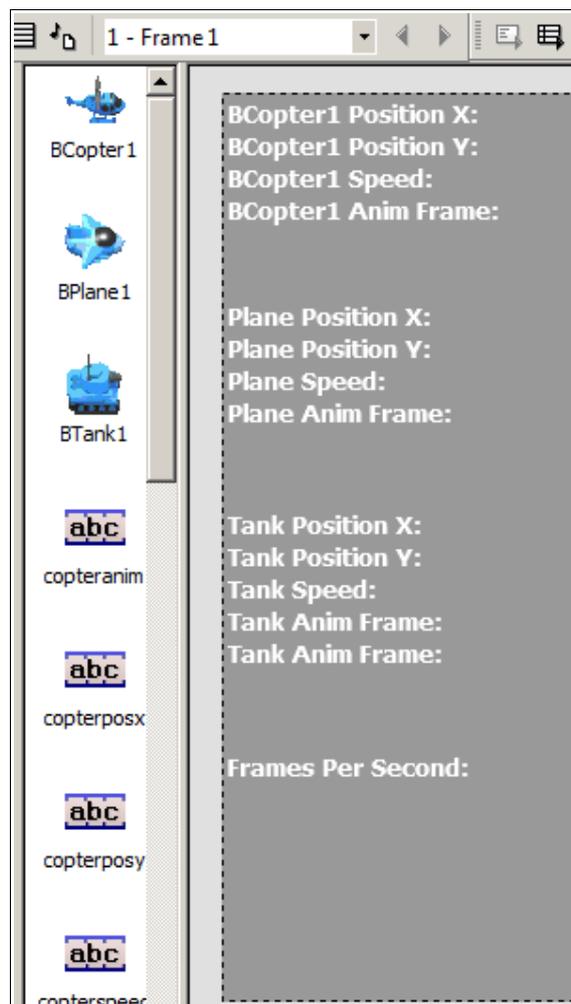
For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

3. CREATING THE DEBUG LAYER

This little 'Tab' we have created with a quick backdrop will be our Debug Tab. On here, we can put anything we want. Now, we will want to display our debug information as strings and/or counters. So let's create some string objects to set the layout of it. Replicate the image shown below:



They are all strings. They are stretched on their bounding box width, to the size of the grey area we put there. We will update these values in real-time to show all of this information in one simple event. Let's head over to the Event Editor...



For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

4. IMPLEMENTING THE EVENTS

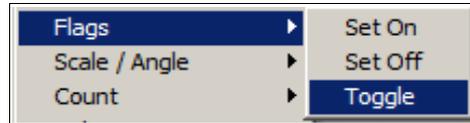
So, first of all, we want to toggle the display of our Debug Layer on or off when we press the tab key. To do this, it's best to Toggle a FLAG. If you haven't learnt about flags in another guide or example, you will learn now. A FLAG can be two states: **On** or **Off**. Kind of like, 0 or 1, True or False. We just test for the state of this FLAG.

So let's create a new event, insert a new condition: **The Keyboard > Upon pressing a key...**



In the prompt dialog, hit the TAB key. This will insert a new event that is displayed:

“Upon Pressing Tab”. So on this condition line, let's change the flag of our player tank. Hover over to the TANK icon, right-click and select: **Flags > Toggle**.



We will toggle Flag 0. So when the expression editor pops up, type in 0 and click OK. Now, let's jump back to the frame editor. We need to insert a new object, insert a new object and select the Layer Object



We need this object in our frame to perform layer actions such as Show/Hide. Jump back to the event editor. Now, we need to test whether a flag is on or off. If the flag is on, we will SHOW the Debug Layer. If the flag is OFF, we will hide the Debug Layer.

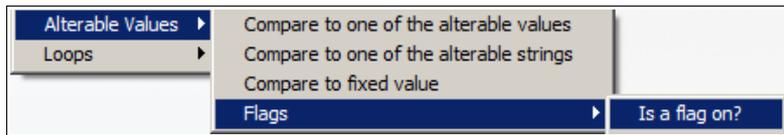
For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

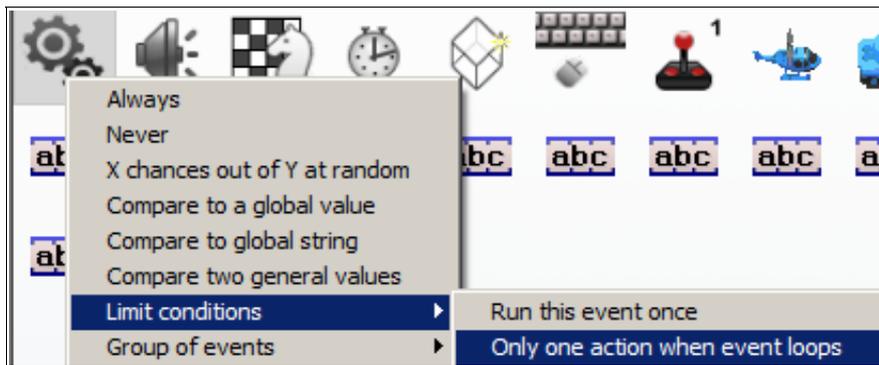
The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

4. IMPLEMENTING THE EVENTS

In the event editor, insert a new condition:



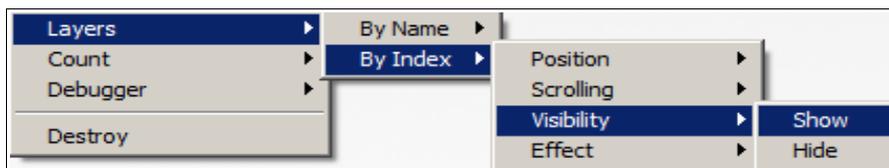
When prompted, type 0 and click ok. Now we need to attach another condition to this existing condition so right-click on that new event we just did and select: **Add a new condition** then select **Special Object > Limit Conditions > Only one action when event loops**.



Your new event should look like this:



This “Only one action when event loops” means it won't keep iterating over that event if it's still in the same state as the last loop. If nothing has changed, what's the point in performing those actions again? On this event line, hover over to the Layer Object, right-click and select: **Layers > By Index > Visibility > Show...**



For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.



4. IMPLEMENTING THE EVENTS

Once you have done this, repeat the above step but instead the condition should be to check if the flag is OFF, then on the actions of the layer object HIDE.

Once you have done this, Hit F8 to run the application and press the TAB key to show the Debug Layer, then press it again to hide the Debug Layer.

So that's perfect. We can now show or hide our Debug Layer at runtime with just the press of a button.

Let's now make our debug strings display some information we want to see at runtime from our objects...

For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

4. IMPLEMENTING THE EVENTS

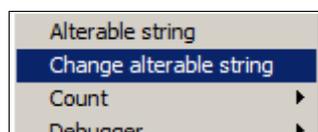
Jump back into the event editor. Now, to display our information, we could do an ALWAYS event, which means 60 times per second (or whatever your FPS is) render all the strings, however, this isn't efficient as when the Debug layer is hidden and we don't need to see any debug information, it will still keep rendering this event.

So, to counter this, instead of ALWAYS, we will use the condition:

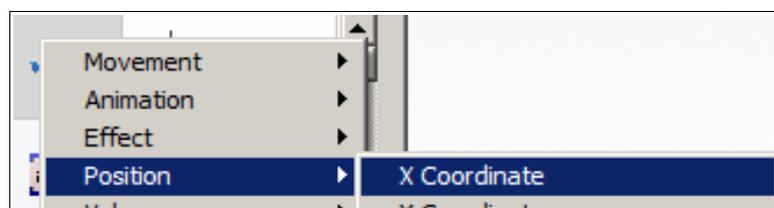


So, we will tell Fusion 2.5 to update the debug strings only when the flag is on, because when the flag is ON, that is the only time the debug layer is being shown to us. We don't use "Only one action when event loops" for this event as we want it to constantly loop (60 times per second) whilst it's showing, so it can keep the debug strings updated.

Now, on this event line, hover across to our first string, which should be the Bcopter1 Position X: string. Right-click and select **Change Alterable String**.



When the expression editor pops up, type in "Bcopter1 Position X: "+Str\$(X("Bcopter1"))... or right-click on the Bcopter1 object and select **Position > X Coordinate**...



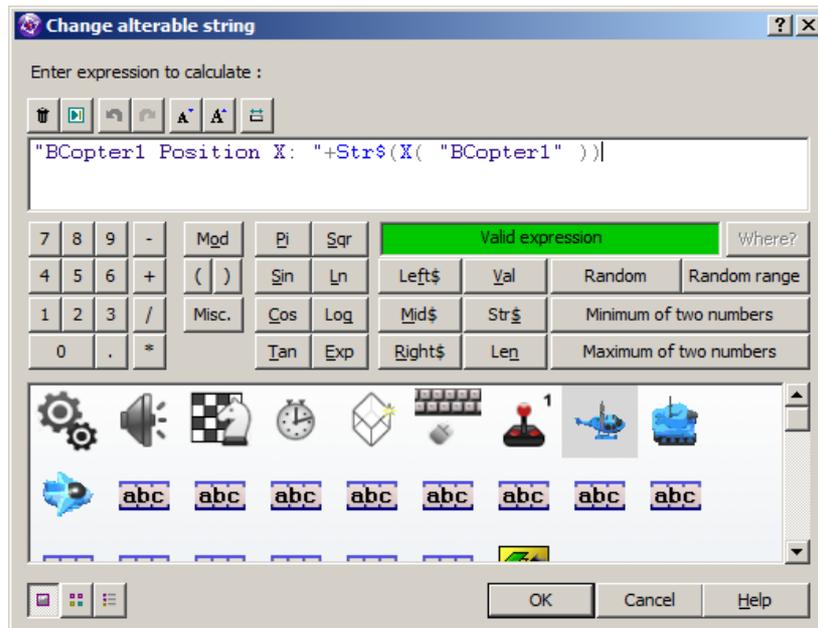
For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

4. IMPLEMENTING THE EVENTS

Your expression should look like this:



Click OK.

Now, repeat this for the next string (on the same event line) but replacing Str\$(X("Bcopter1")) with (Y("Bcopter1")) as we want the Y position for the next one. So your final expression should look like this:

"Bcopter1 Position Y: "+Str\$(Y("Bcopter1"))

Once that is done, Hit F8 on your keyboard to run the application. Press TAB to display the debug layer and you should see those strings updating 60 times per second. They are telling you the exact X and Y coordinates of the Bcopter1 object all the time. Handy feature? Sure. Let's implement the next few debug strings...

N.B The Str\$() command allows Fusion to convert a number to a string. Seeing as though the X and Y values of the moving object are numbers, we need the Str\$() command to convert this number (value) into a string to be displayed as text.

For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.

5. TESTING THE DEBUG SYSTEM

The next one down is Bcopter1 Speed, so hover over to the bcopter1 speed string, right-click and again, select “Change Alterable String”. In the expression editor, type in “Bcopter1 Speed: “+Str\$(Speed(“BCopter1”)) like so...



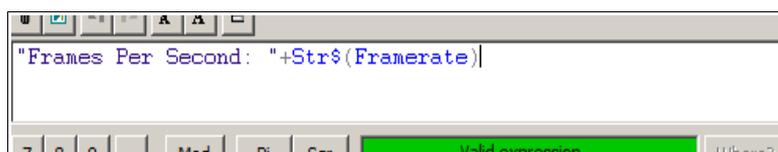
Click OK. Again, move to the next string, right-click and select “Change Alterable String” then type in: “Bcopter1 Anim Frame: “+Str\$(Image(“BCopter1”)) like so...



Great. That's all 4 debug strings we setup regarding the Bcopter1 object. Hit F8 on your keyboard to run the application again. Now press TAB to show the Debug Layer and you will see all 4 values we wanted to see for the Bcopter1 object showing in real time.

Once you have done this, repeat the above process for the other objects, but keep them on the same event line. Just move through each debug string and type into the expression editor what values you want to show. Don't forget you will need to use the Str\$() command when you want to convert a number (value) to a string.

Finally, remember, we had the last string as “Frames Per Second” ? Let's implement that. This is going to show us at how many frames per second our application is running at (in realtime). On the same event line, hover over to the frames per second (fps) string, right-click and select “Change Alterable String”. In the expression editor simply type: “Frames Per Second: “+Str\$(Framerate)



Hit F8 again, to see all of this in action.

For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

6. FINAL NOTES

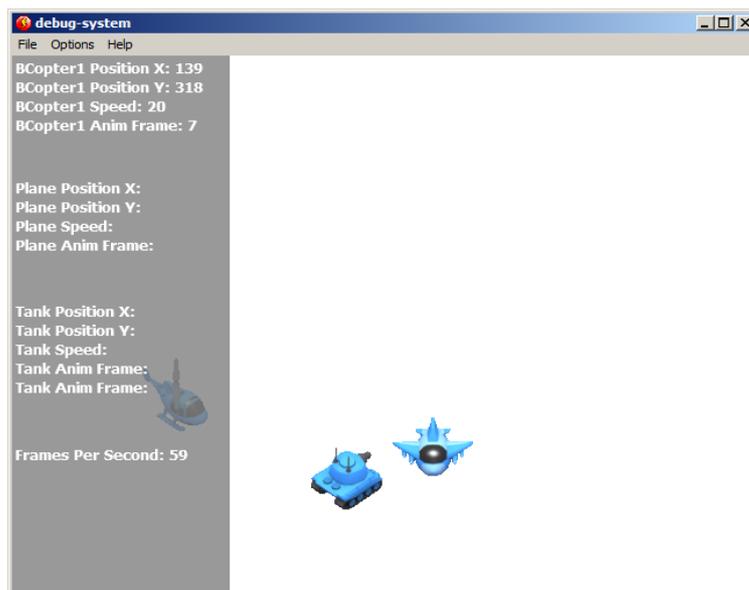
Hopefully above, you have learnt how to create your very own customised debug system or 'debug layer' in Fusion 2.5. You can totally control and completely customise every single aspect of what appears on there. You can get a lot of information about any object at runtime, literally, any object. Even objects like INI, Progress Bars or practically any other object that has a list of expressions you can retrieve.

Again, a debug system may not be necessary for every single game you develop, but for some medium-large size projects it probably will be.

It's handy if you're testing a large evented game you're working on as you will want to see what is happening all the time.

The second big plus to using a debug system is, if you are having a problem with something you've done and cannot understand why, with a debug system displaying any information you want, you can trace, track and locate problem events by seeing what is happening all the time.

Do continue to expand upon this guide. On the forum, the thread for this guide features a downloadable ZIP file where you can download the MFA I created whilst writing this guide, for a better insight if you need it.



For more information, tutorials, examples and walk-thrus visit the website.

Website: <http://www.Clickteam.com> – **Community:** <http://community.clickteam.com>

The Clickteam Fusion logo and all content copyright (c) 2013 Clickteam LLC. Unless otherwise stated.