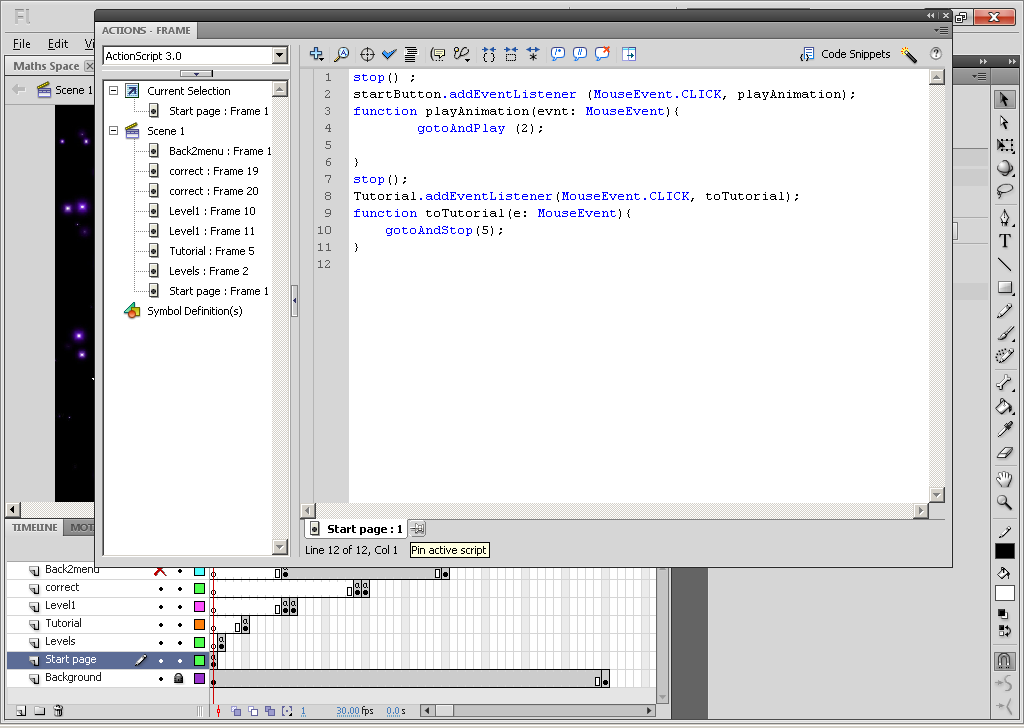
Use a variety of testing tools

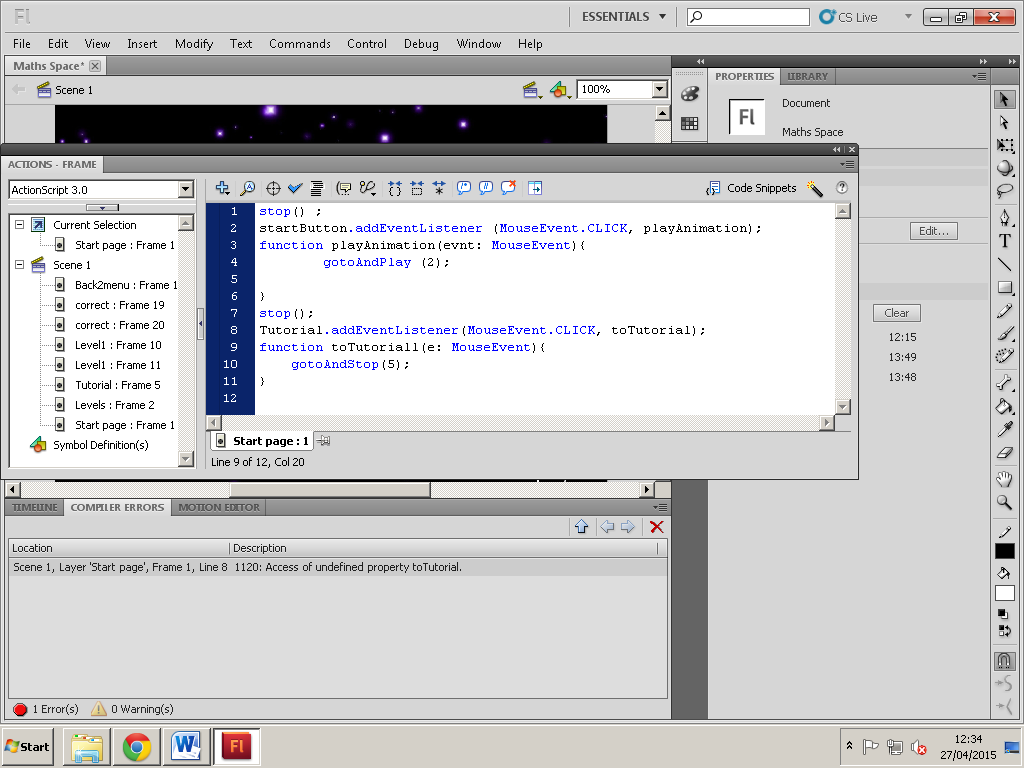
For testing my game, I will be using a variety of tools to do so.

Error Message

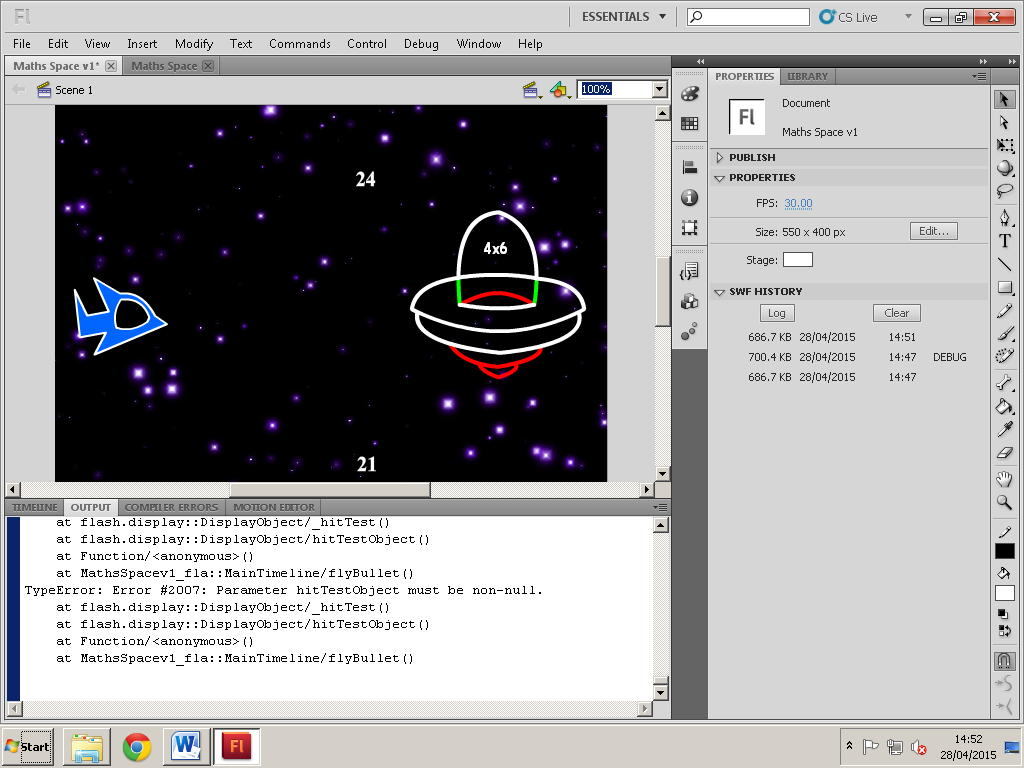
These are some codes from my game, which when a user presses the start button, it will direct the player to another frame.

They are currently correct.

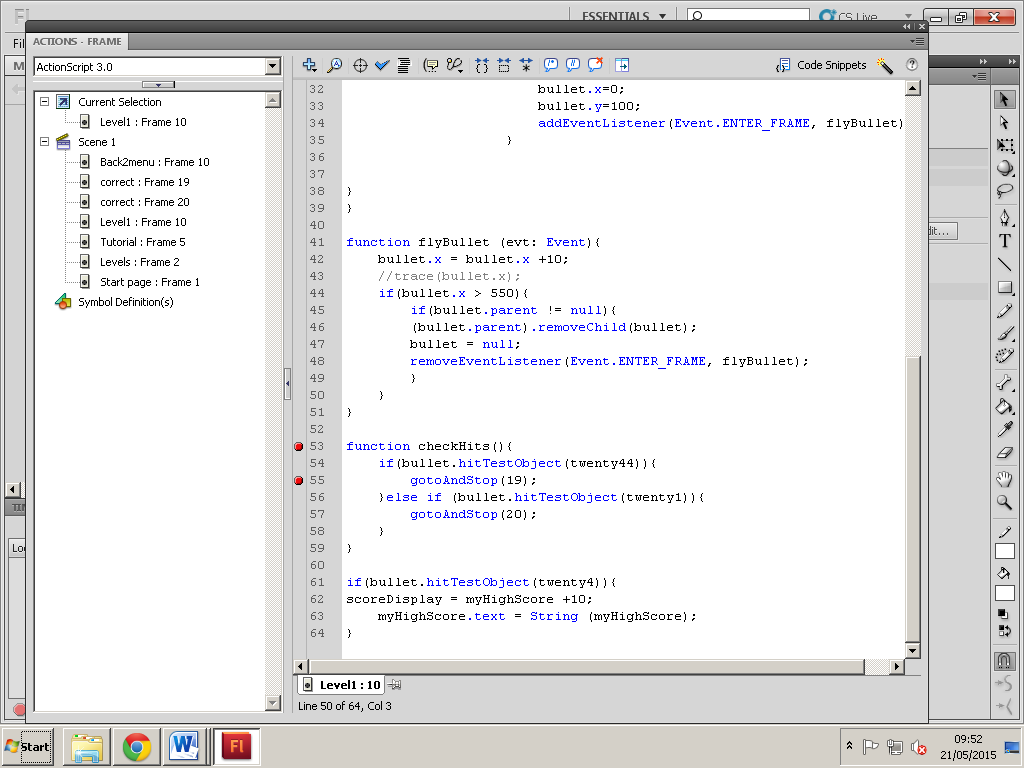
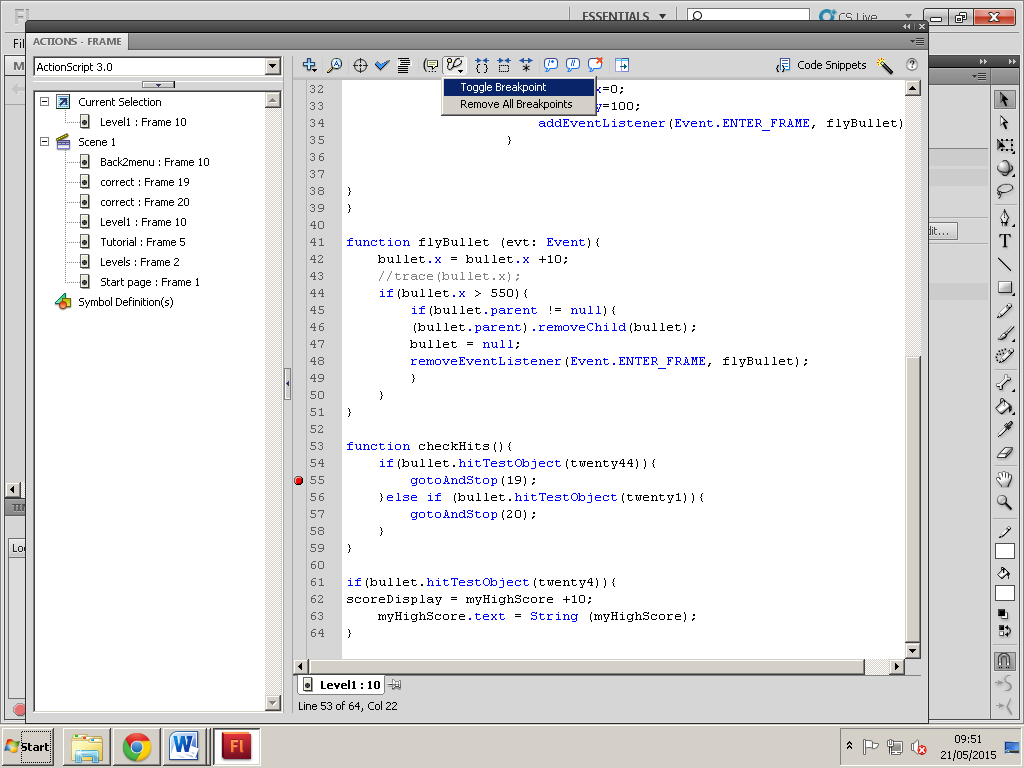
If I am to have an error in my codes, for example I have typed in a different a mistake in the handler, an error message will appear letting me know that something has gone wrong in the codes.



Debugging  
Debugging is used to find exactly where certain errors are in your flash file.

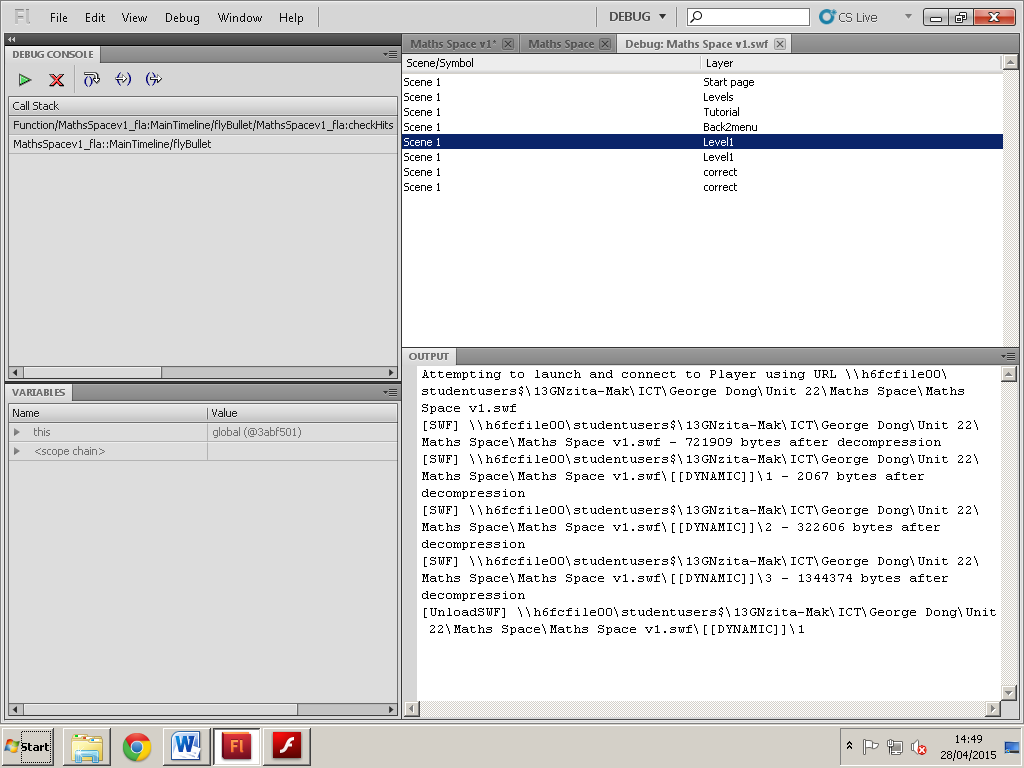


Whenever my flash file is ran, output will show up letting me know that there is a problem with the hitTestObject, that it is appearing to be “null”. This is a runtime error that occurs, meaning that although the codes are correct, I have written the code of the hitTestObject to be null, meaning that I have to change that in order to get it to be non-null.

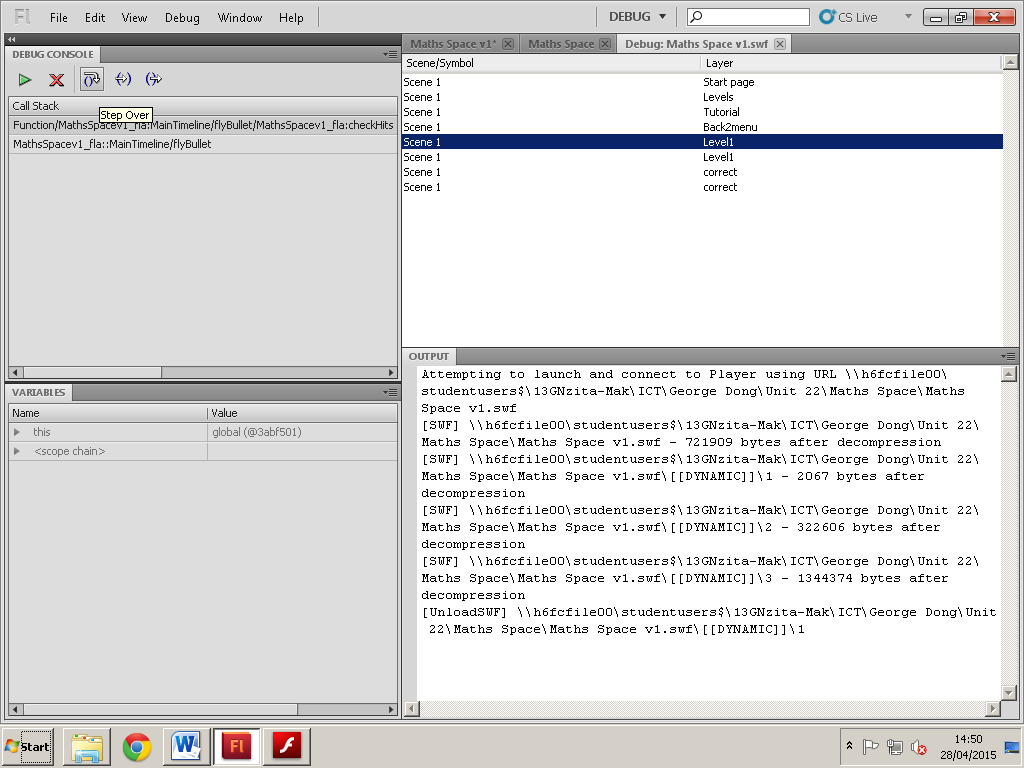


As there is an error that has been detected in my game, by opening up the coding area, before debugging I click on the debug sign, and click “Toggle Breakpoint”, and what happens is that it displays a little red dot on where the errors are.

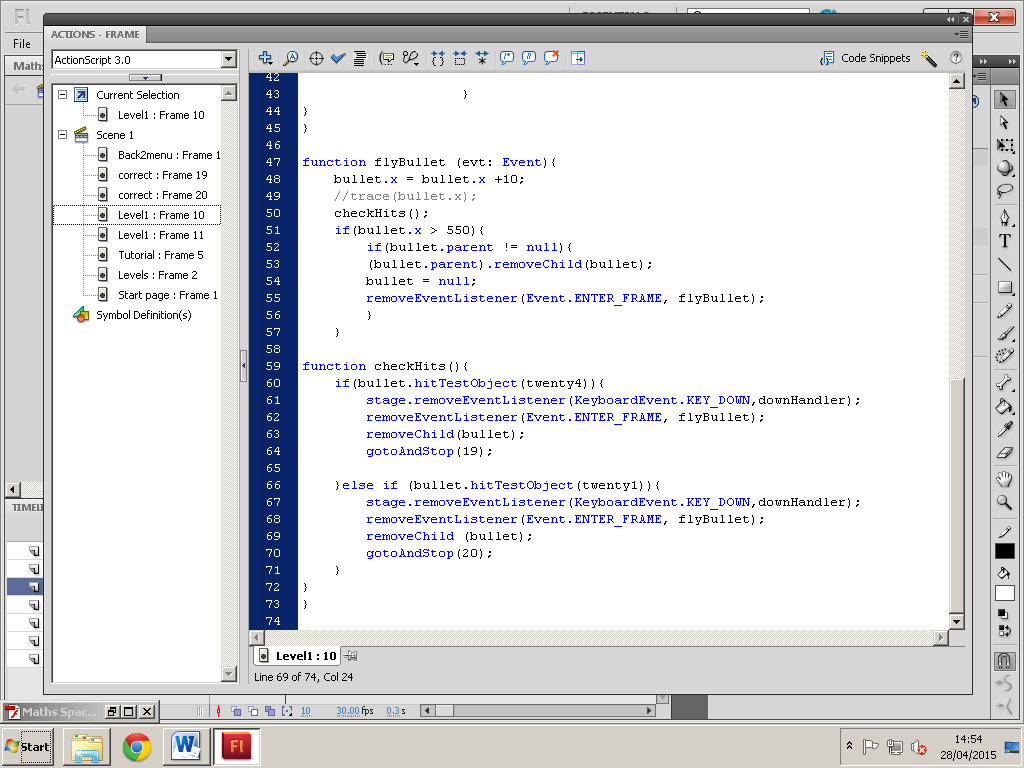
Breakpoints interrupt the code, when the game is ran it will stop and freeze at the point of the breakpoint so that you can either step in, out or over the code to edit out the mistakes that you have made.



So for instance, because I have a problem in the layer names “Level1”, now that I have set a breakpoint, I open up the debugging tool to view and edit the errors.



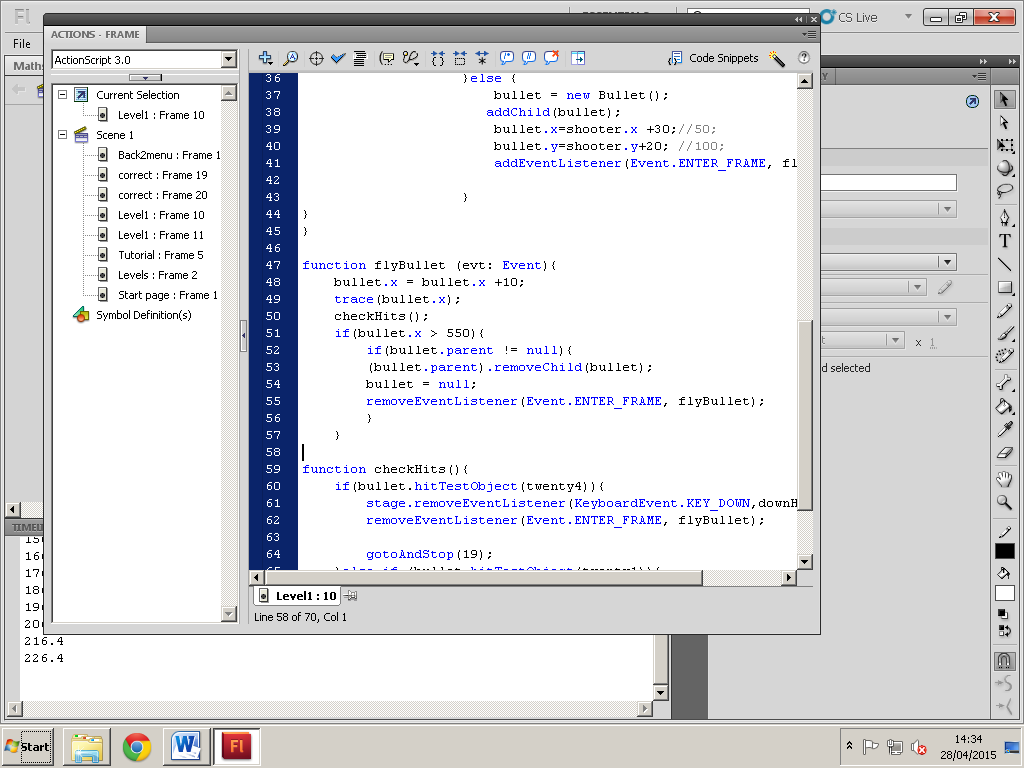
By clicking step over, it tests if it can step over the line of codes, and if there is no problem with that line, then it can be stepped over, whereas if there is a problem with that line then it can’t and I would have to do something about it in order to fix it.

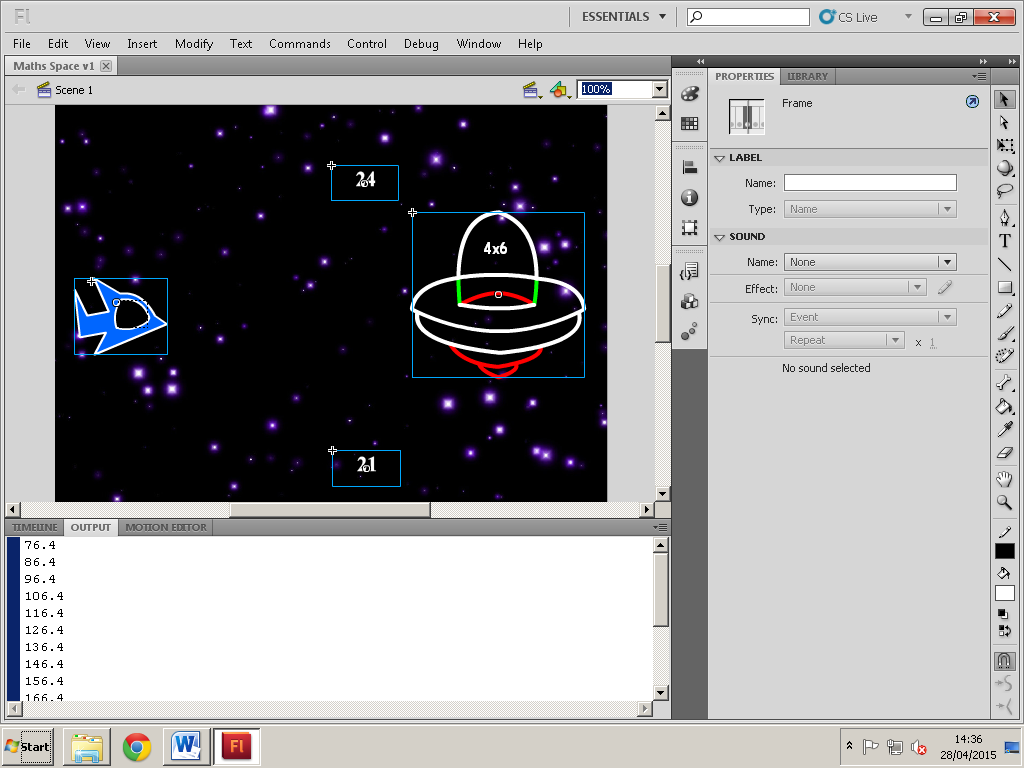


These are the codes in which I have to fix in order for the output to be fine, and in order to fix my game properly.

Trace () function

The trace function prints out feedback of the games output onto the screen.  
In this case what I have decided to trace is the bullet that is being shot out of its shooter.  
I have written in the code “trace (bullet.x) +10;” to trace exactly how far out the bullet goes in the game when it has been shot, and once shot, it prints out the output below





These are the outputs shown when the bullet is shot; giving the x coordinates of the bullets for each and every frame it goes.