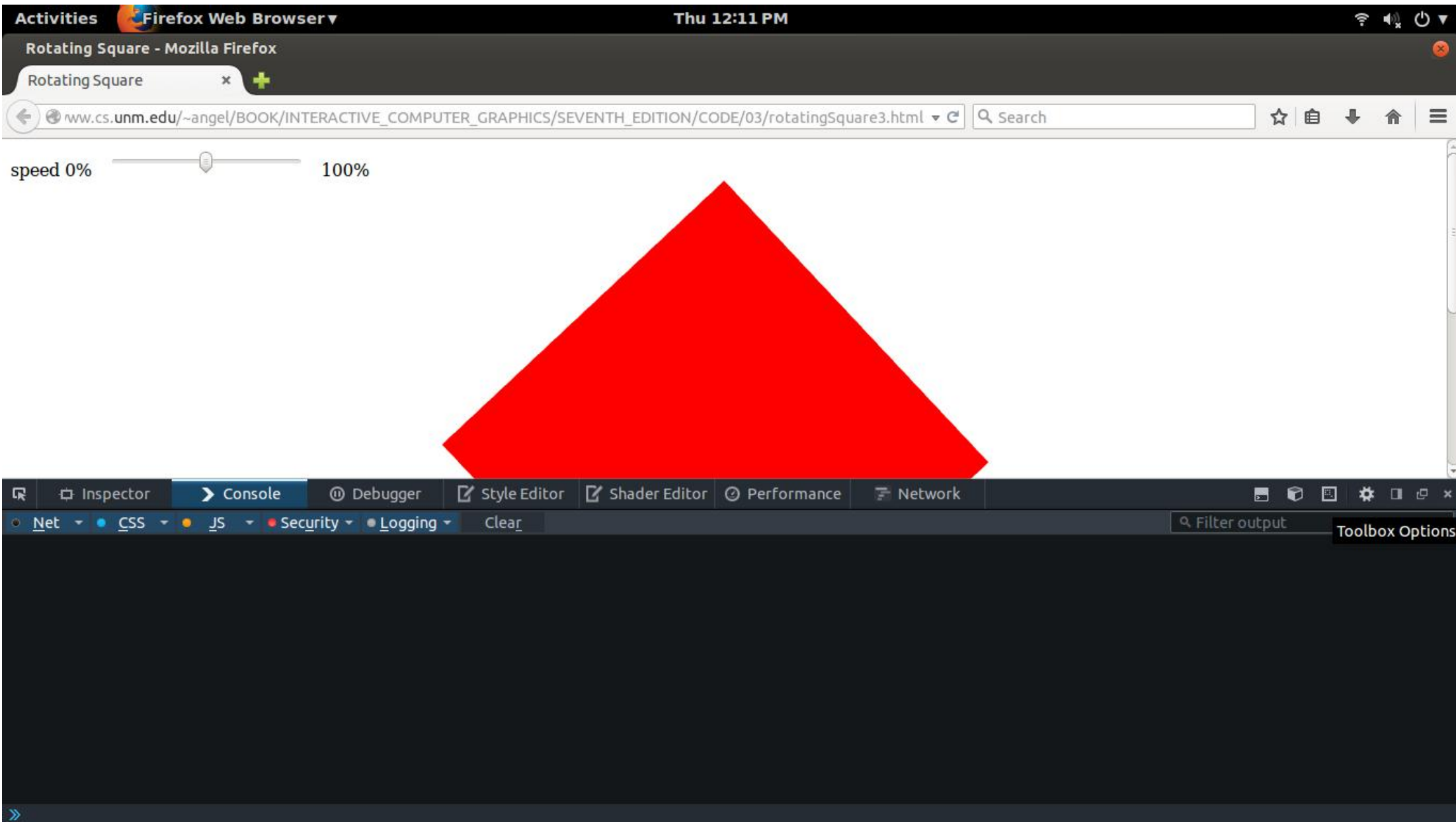
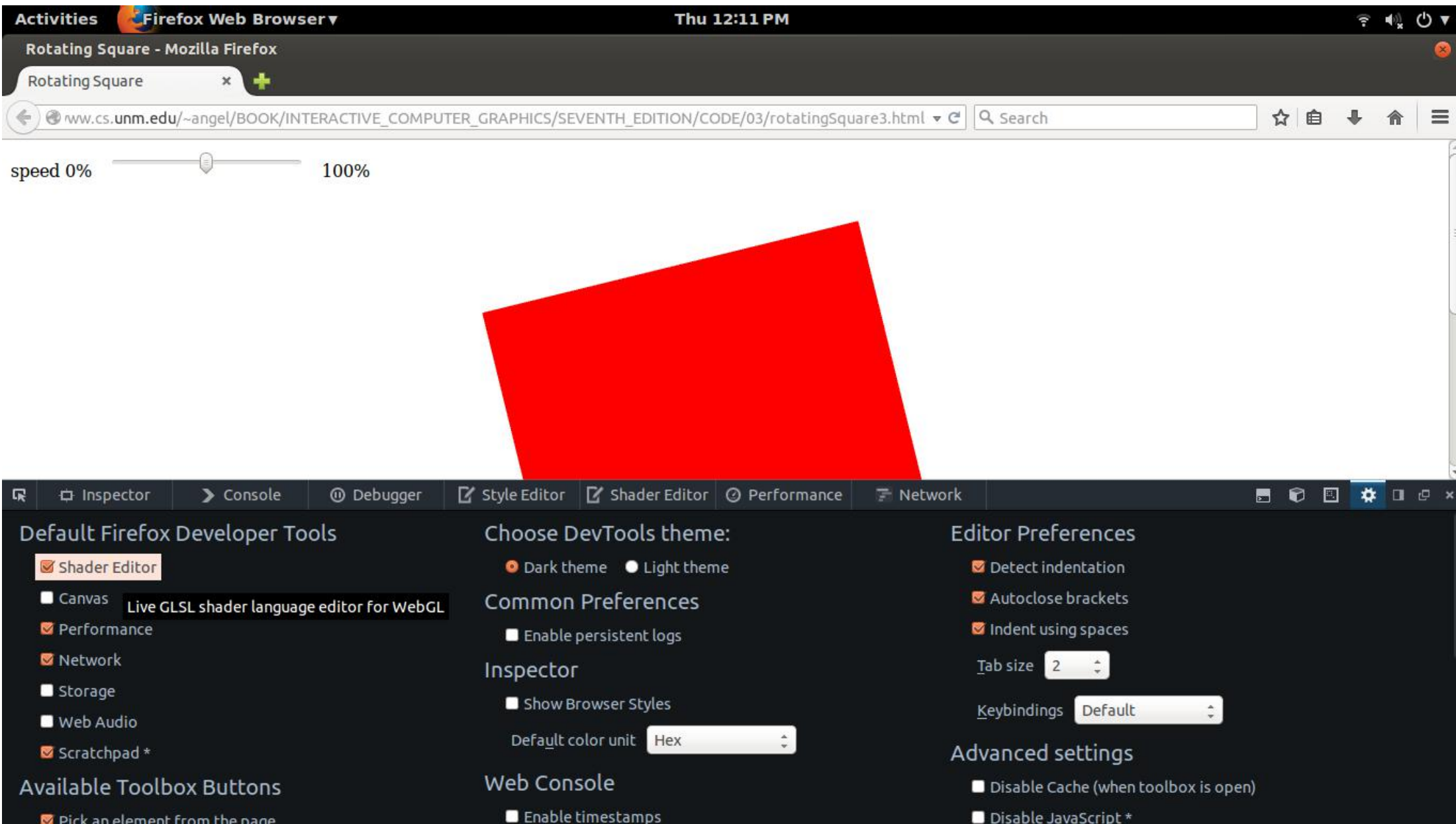


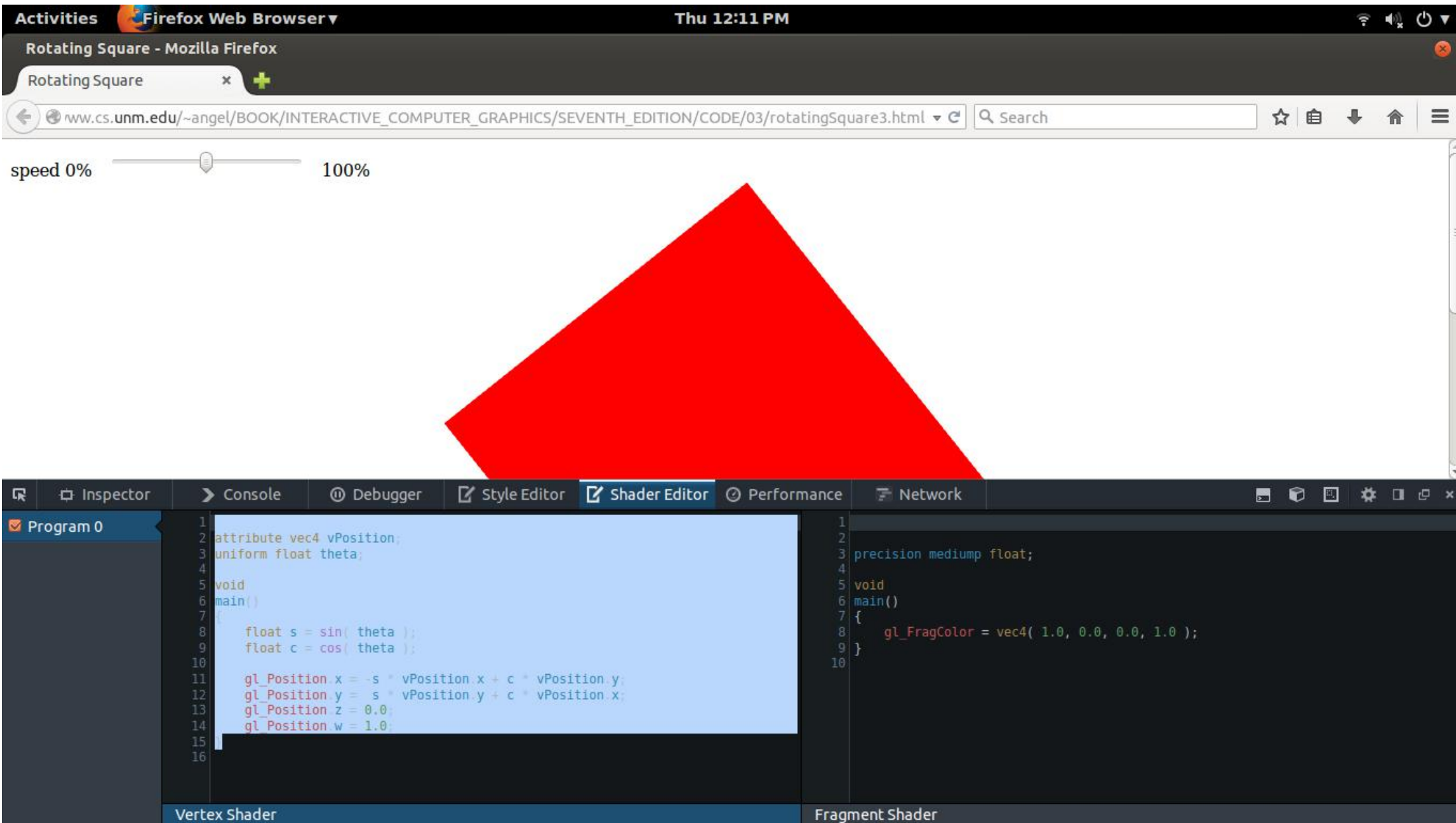
Open Firefox and in the tools menu you will find the Web Console Option. The Web Console is the main tool used by web developers to use Firefox as a dev environment.



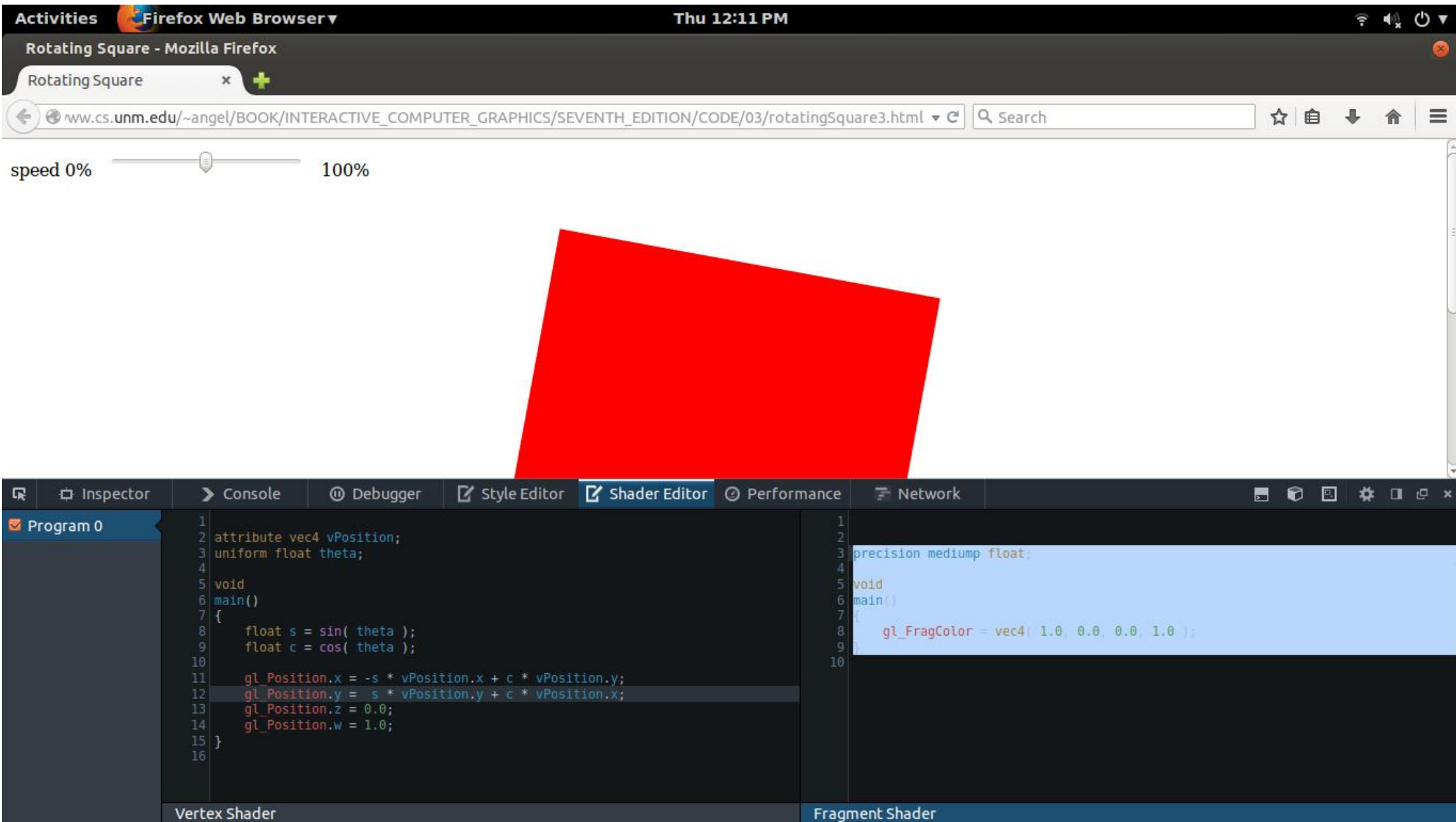
The first thing needed in order to do WebGL development in Firefox is to enable the Shader Editor. Click the gearbox on the right hand side of the Web Console Window to go to the options.



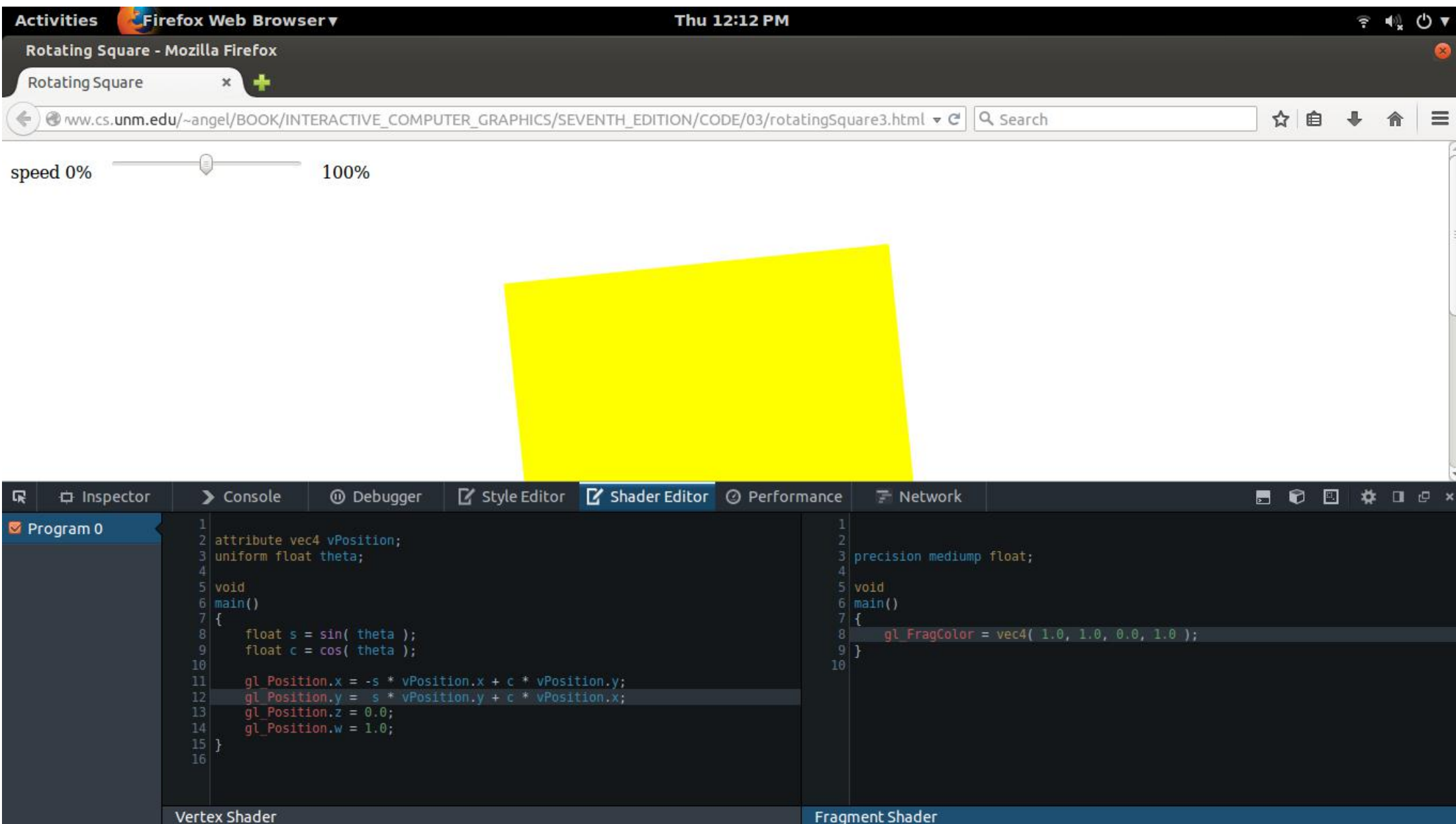
On the left you will find the option to enable the Shader Editor.



This is the shader editor panel. The left sub window is the vertex shader program that is currently active. If there are multiple programs loaded in the GL context, they will appear in the left had list.

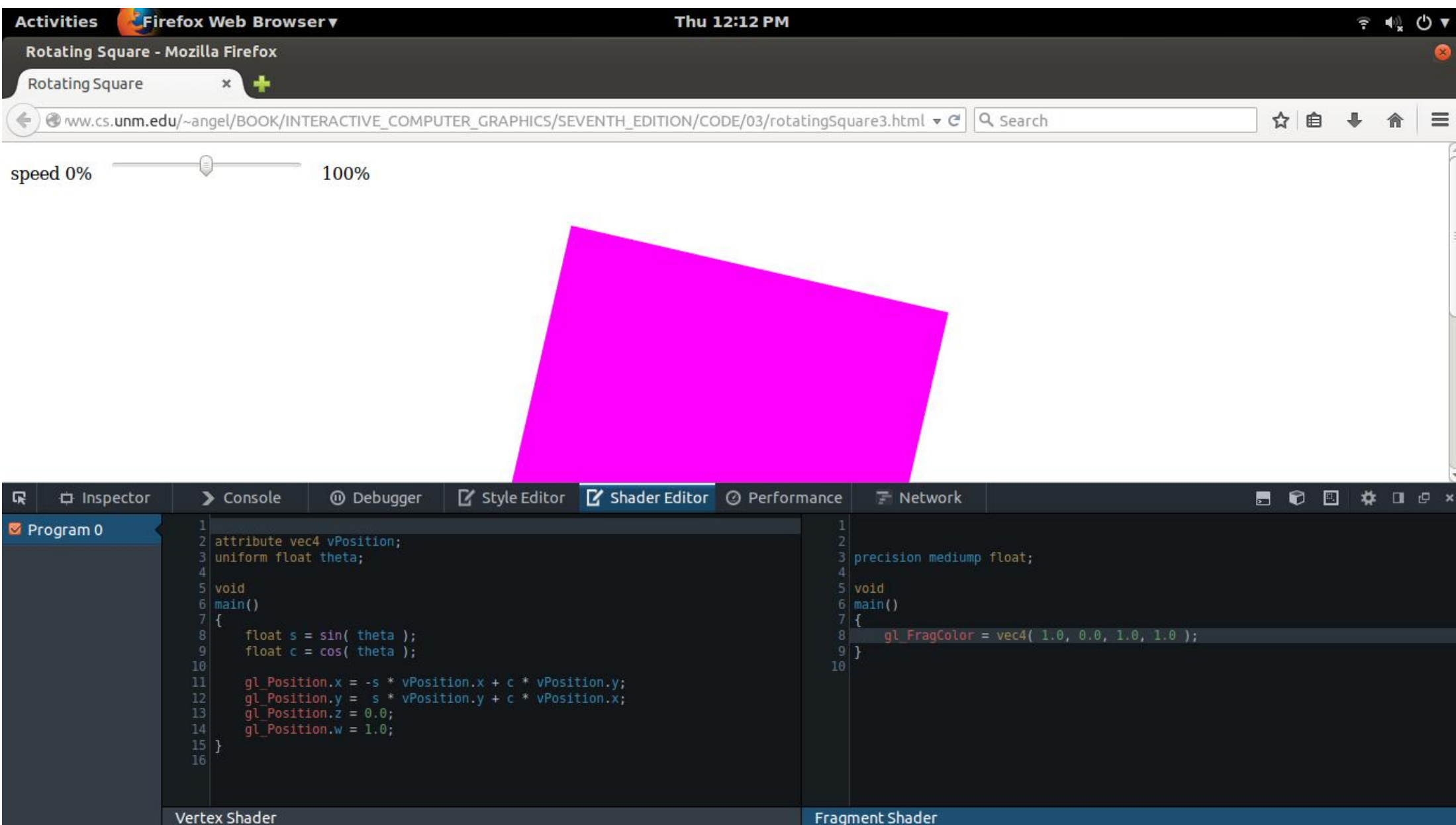


The right subwindow contains the fragment shader program currently in use. Both the Fragment and the Vertex shader are dynamically editable in these windows.

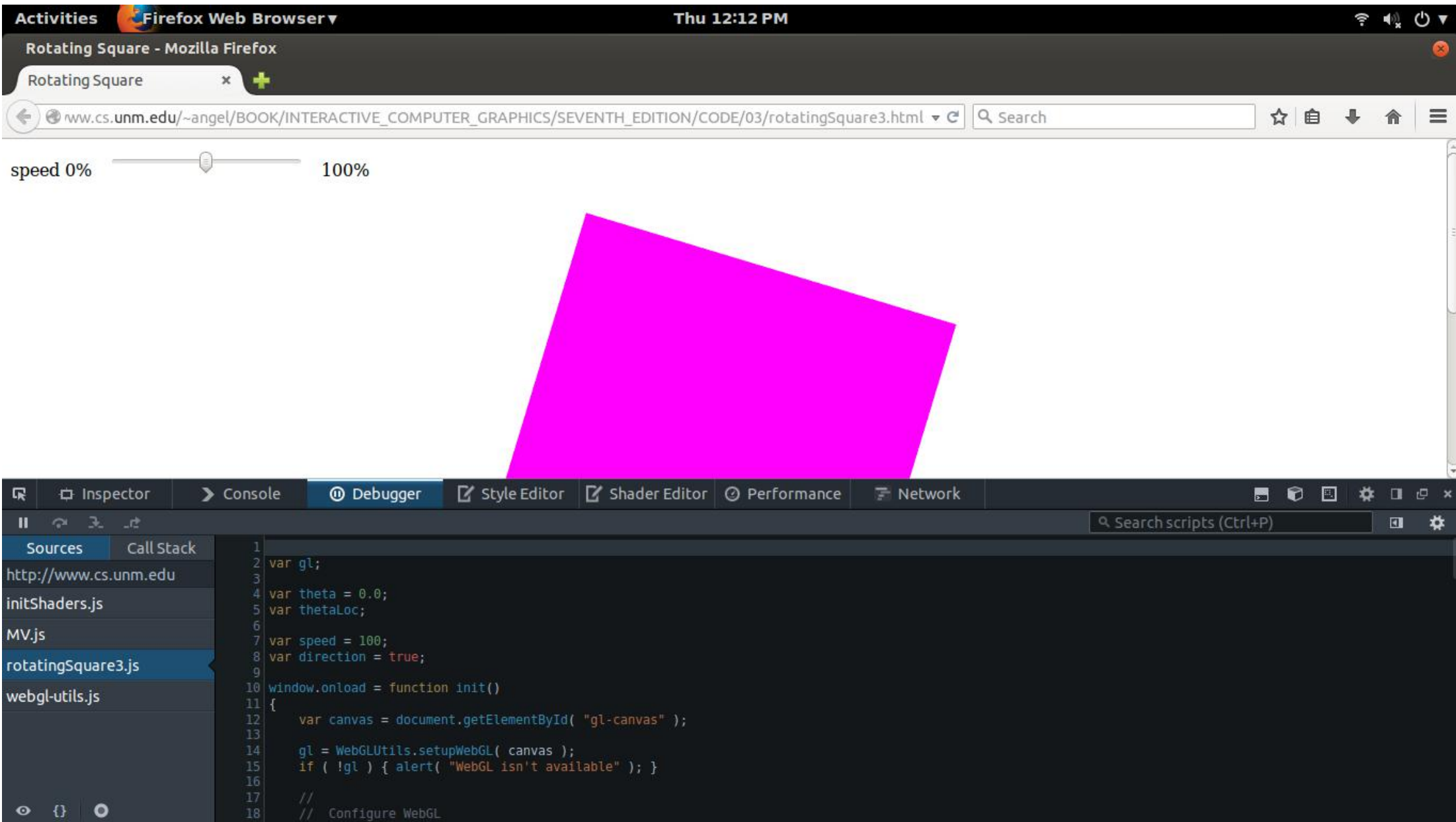


Here I've changed `gl_FragColor = vec4(1.0, 1.0, 0.0, 1.0);` and the effect is to change the color of the square on screen to the edited color, yellow.



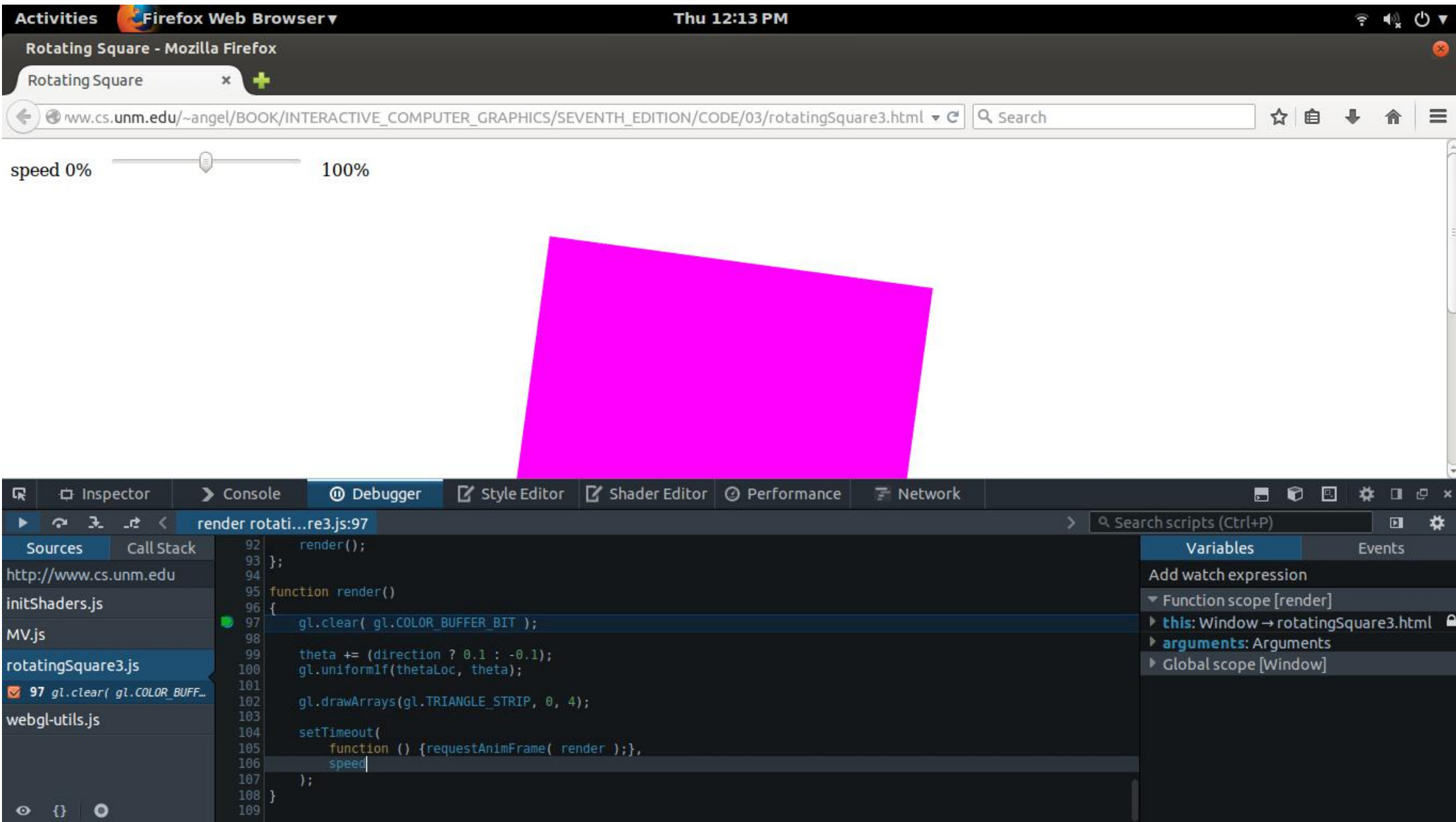


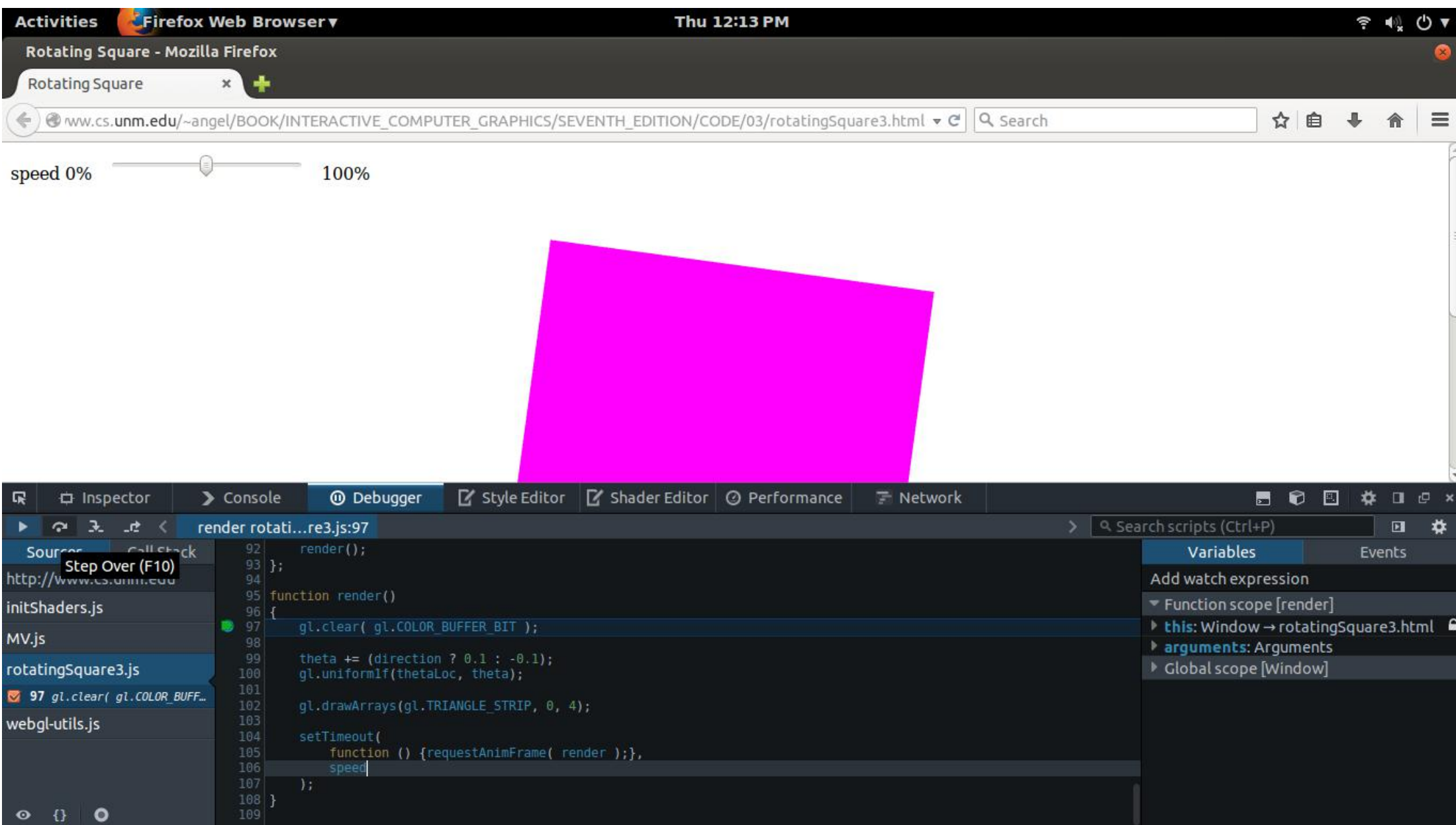
Here I've changed `gl_FragColor = vec4(1.0, 0.0, 1.0, 1.0);` and the effect is to change the color of the square on screen to the edited color, magenta.



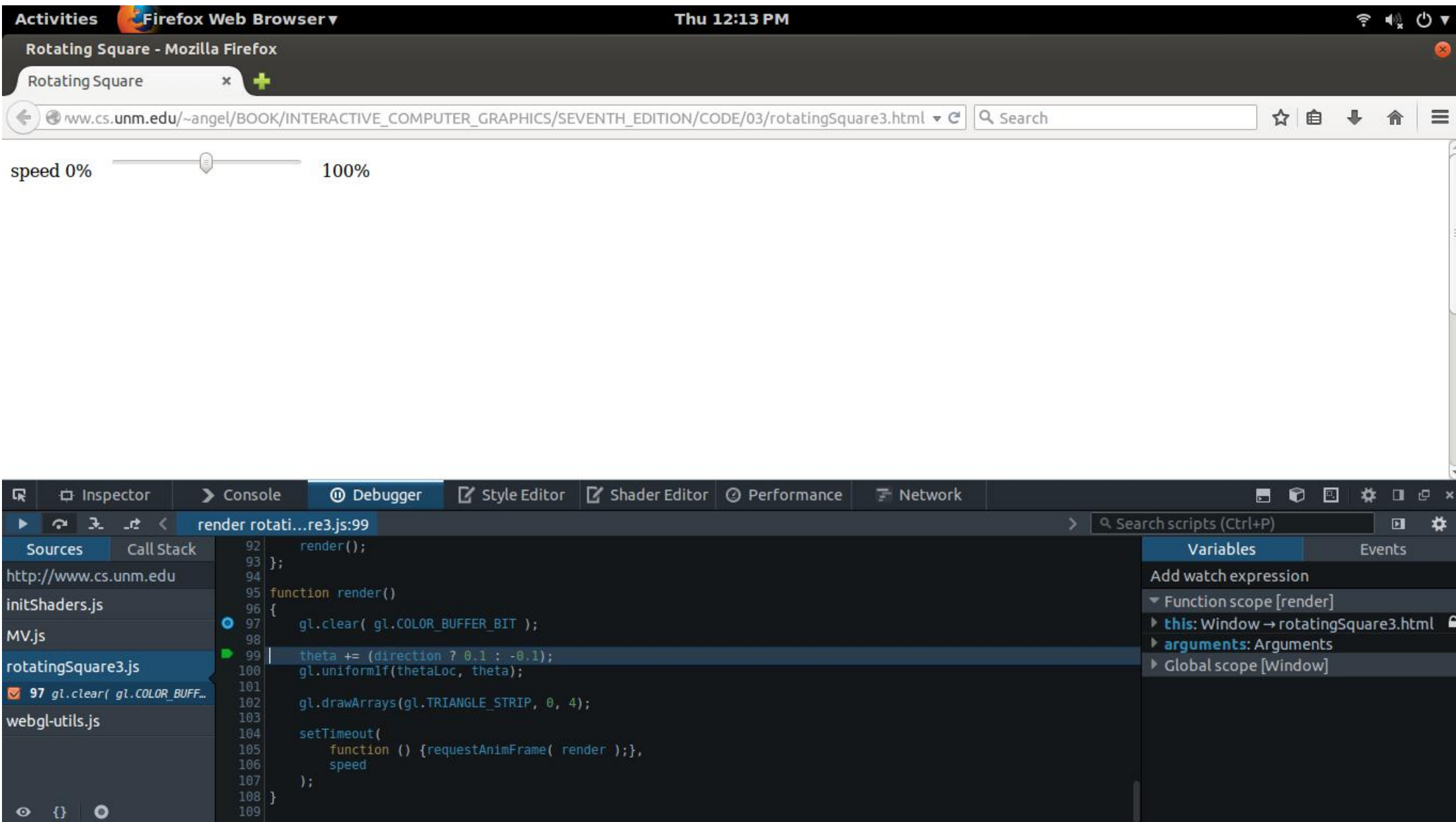
This is the debugger panel. It contains the source of any scripts loaded into the current context. These scripts are not editable.



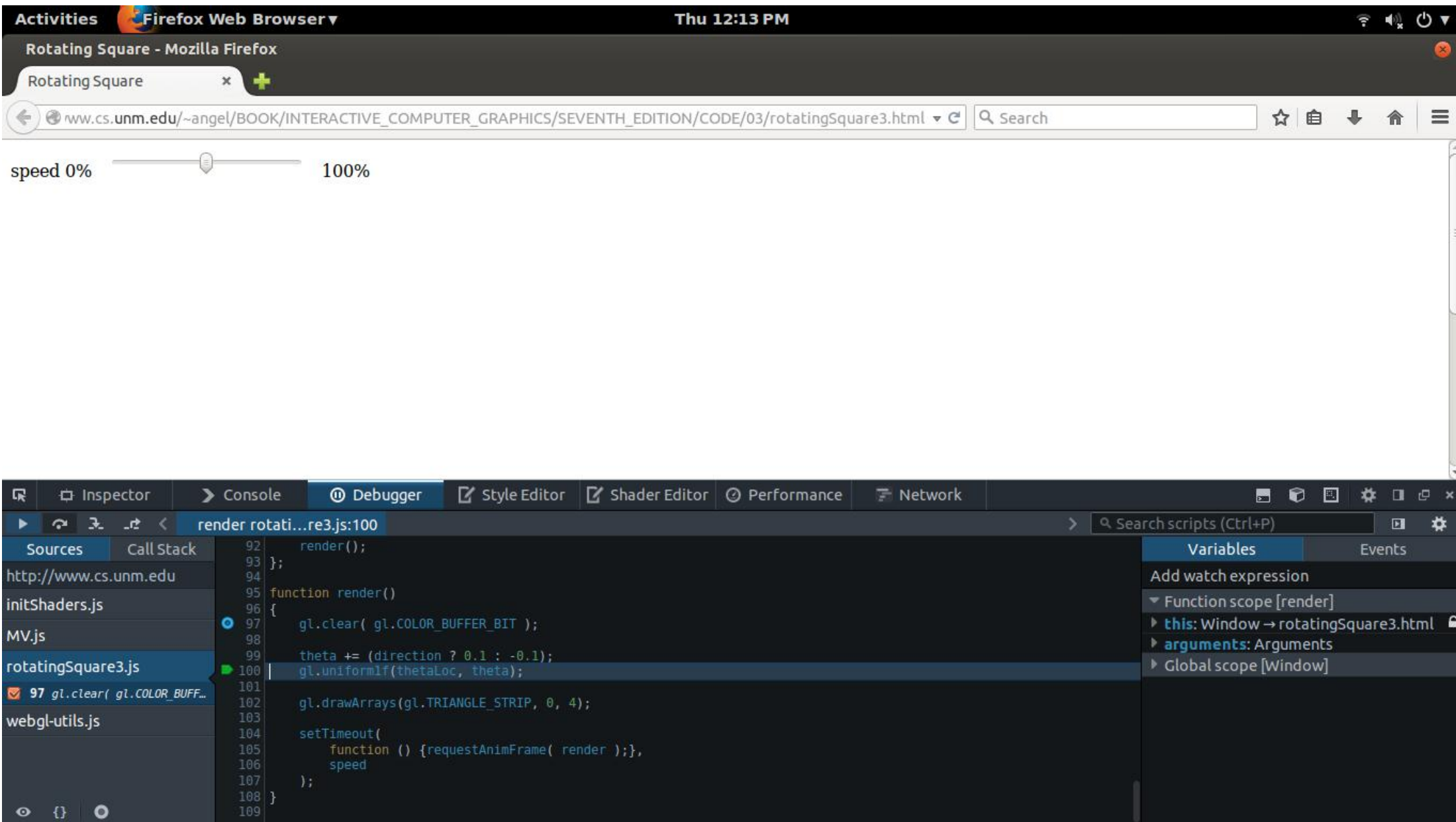




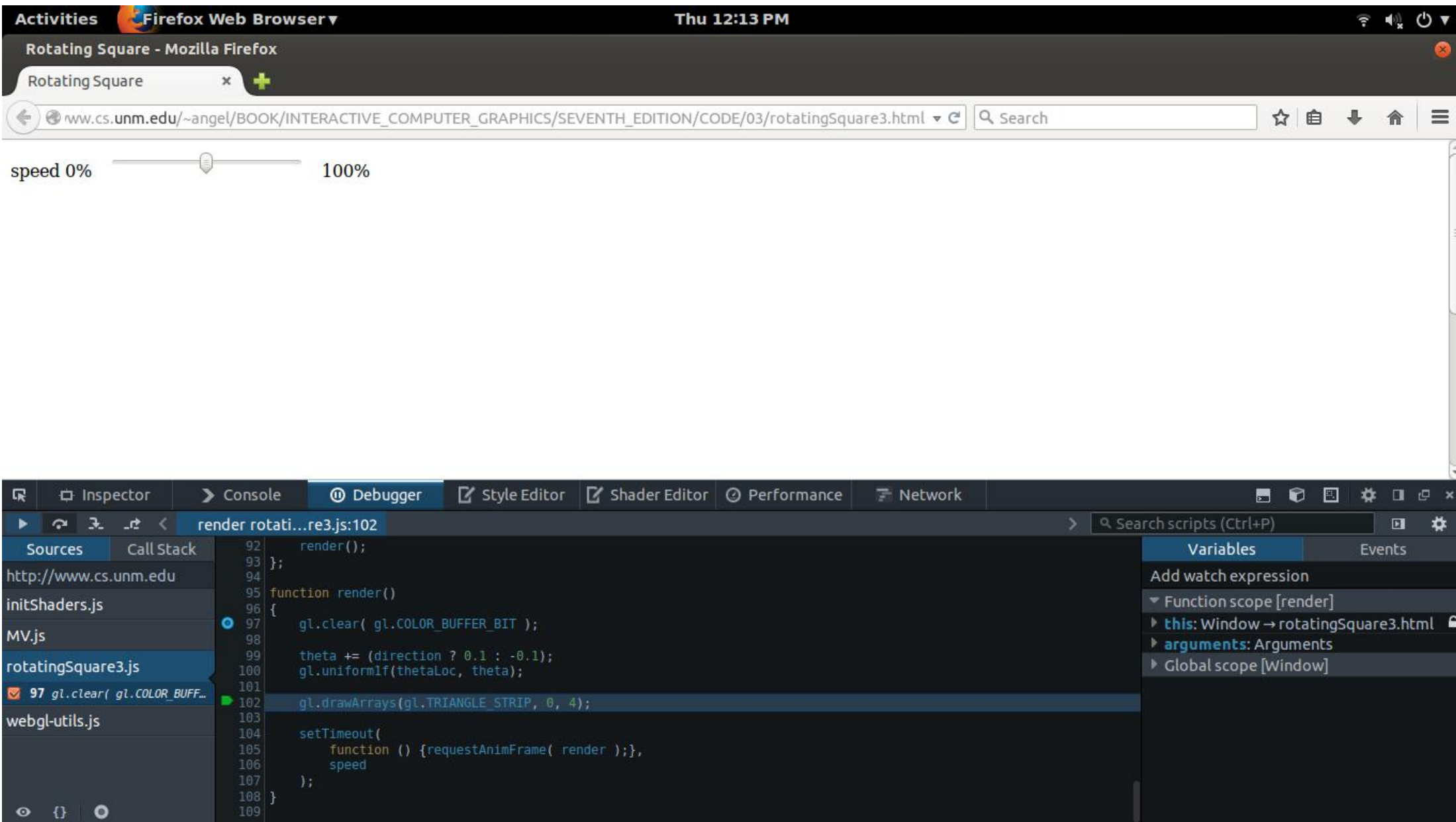
There are buttons on the top left of the panel to step over, into and through each line as is typical of most IDEs.



Step over

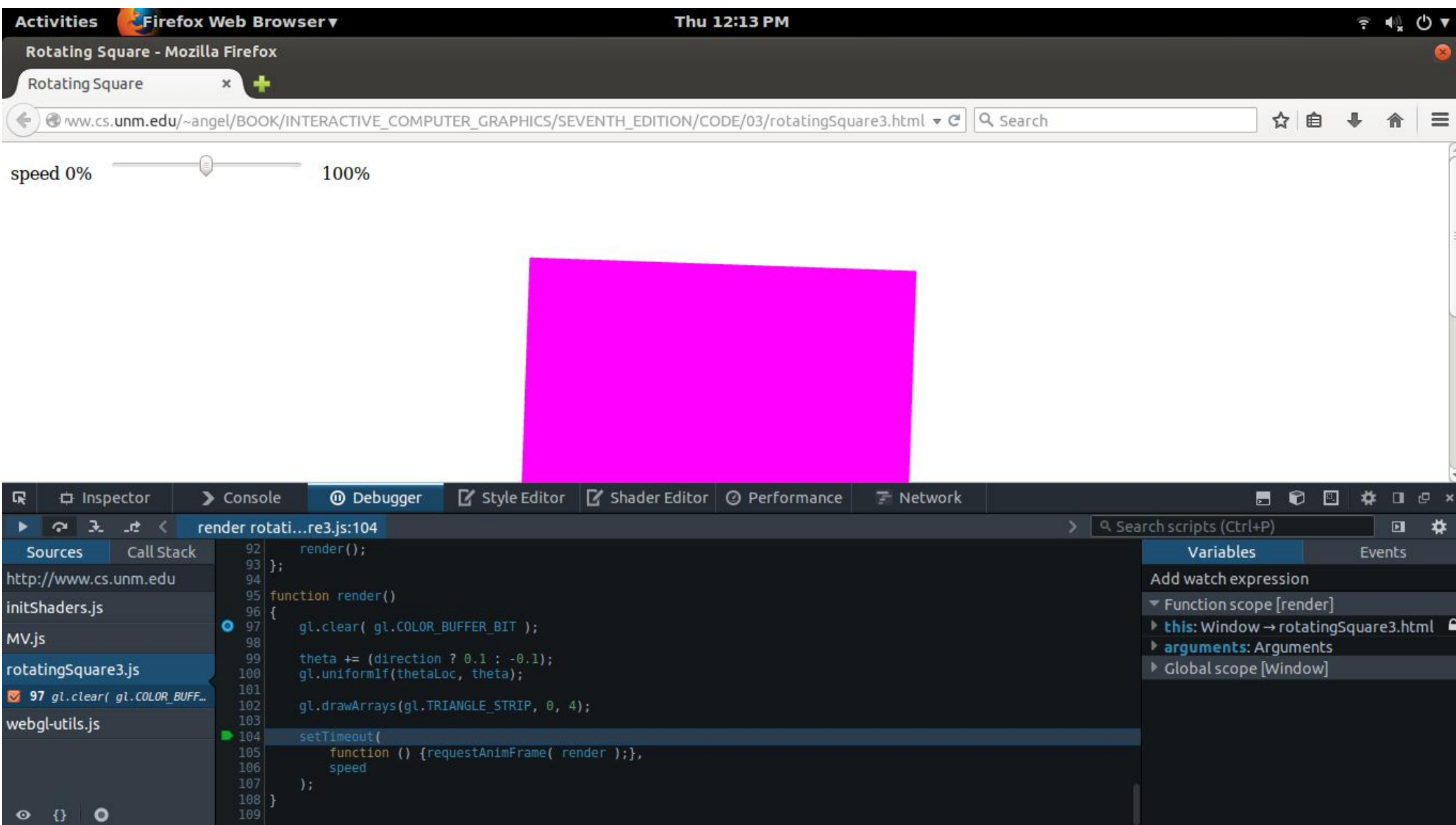


Step Over



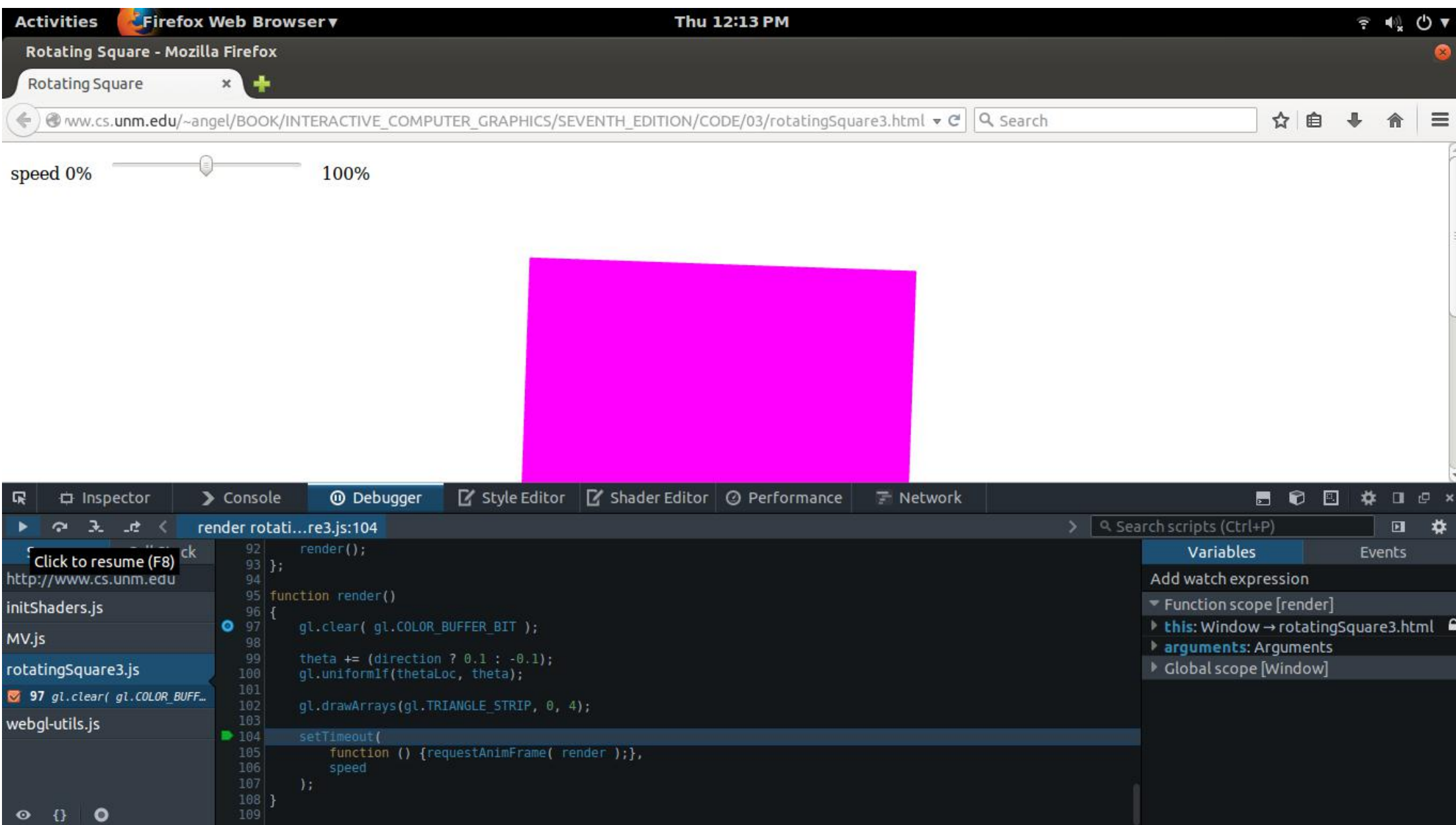
Step Over



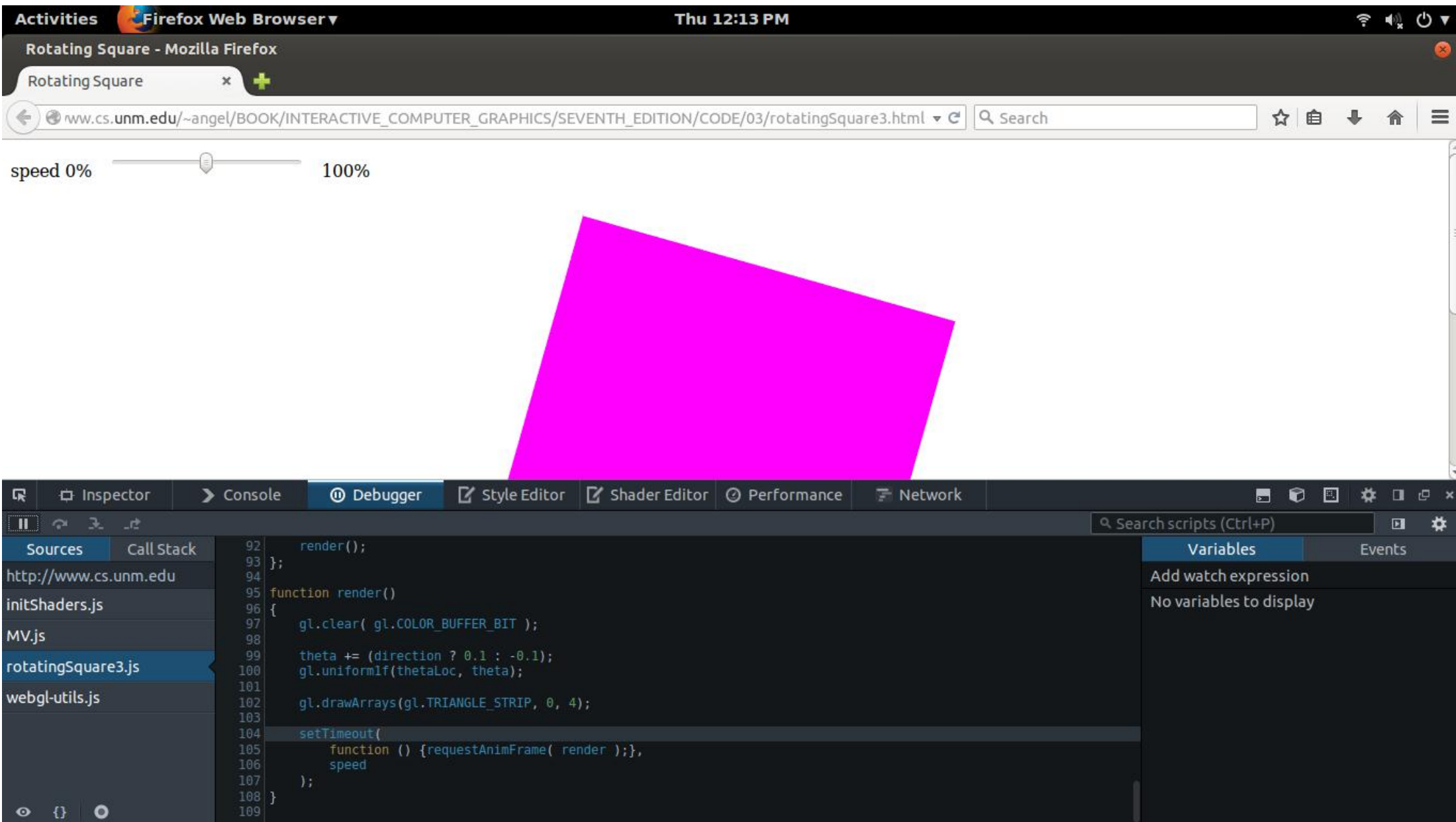


The last line we stepped over was the draw call, and as you can see, the square was rendered to the canvas.

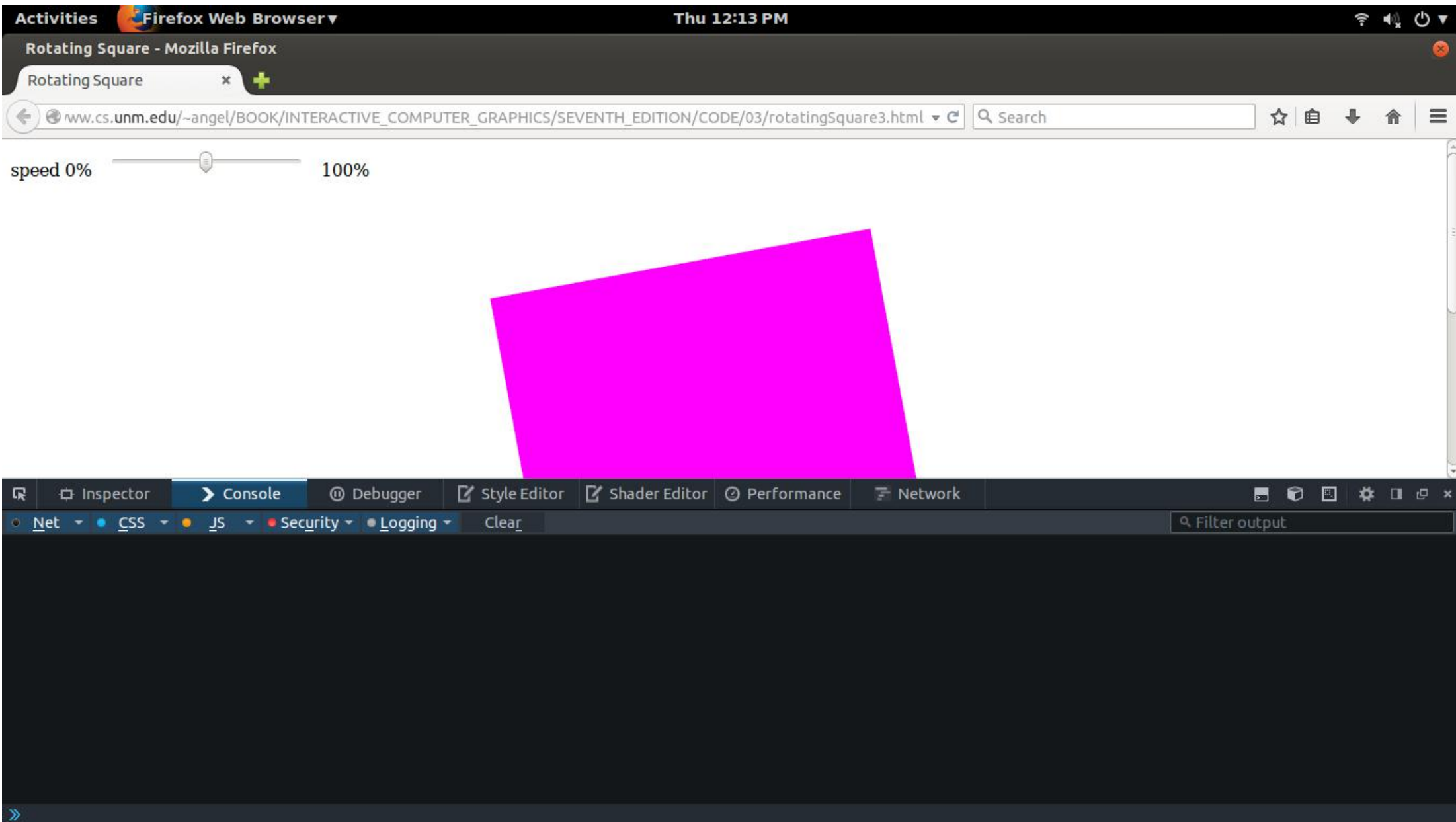




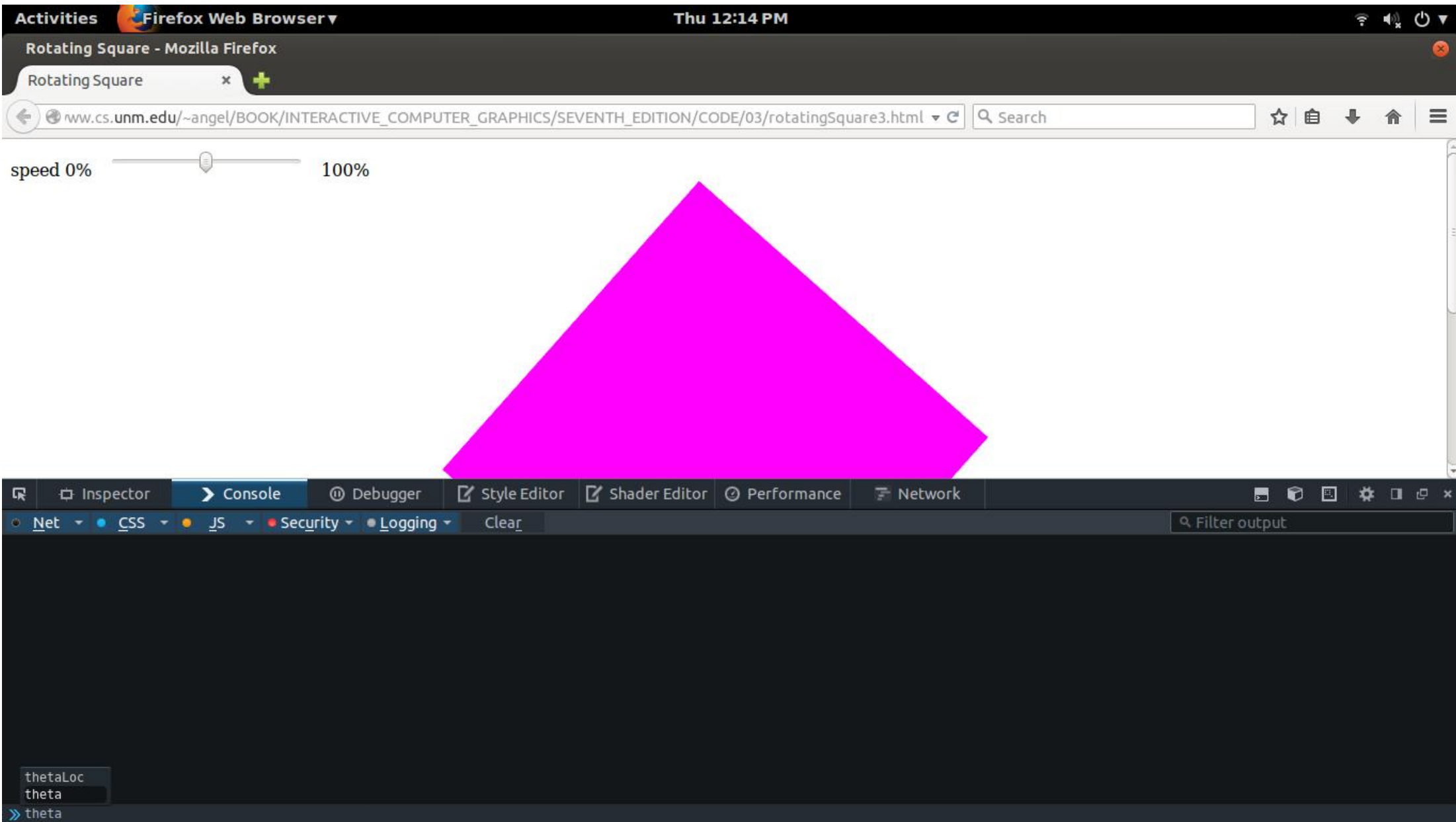
The play button in the far left corner let's you "continue" or resume interpreting the code until the next breakpoint.



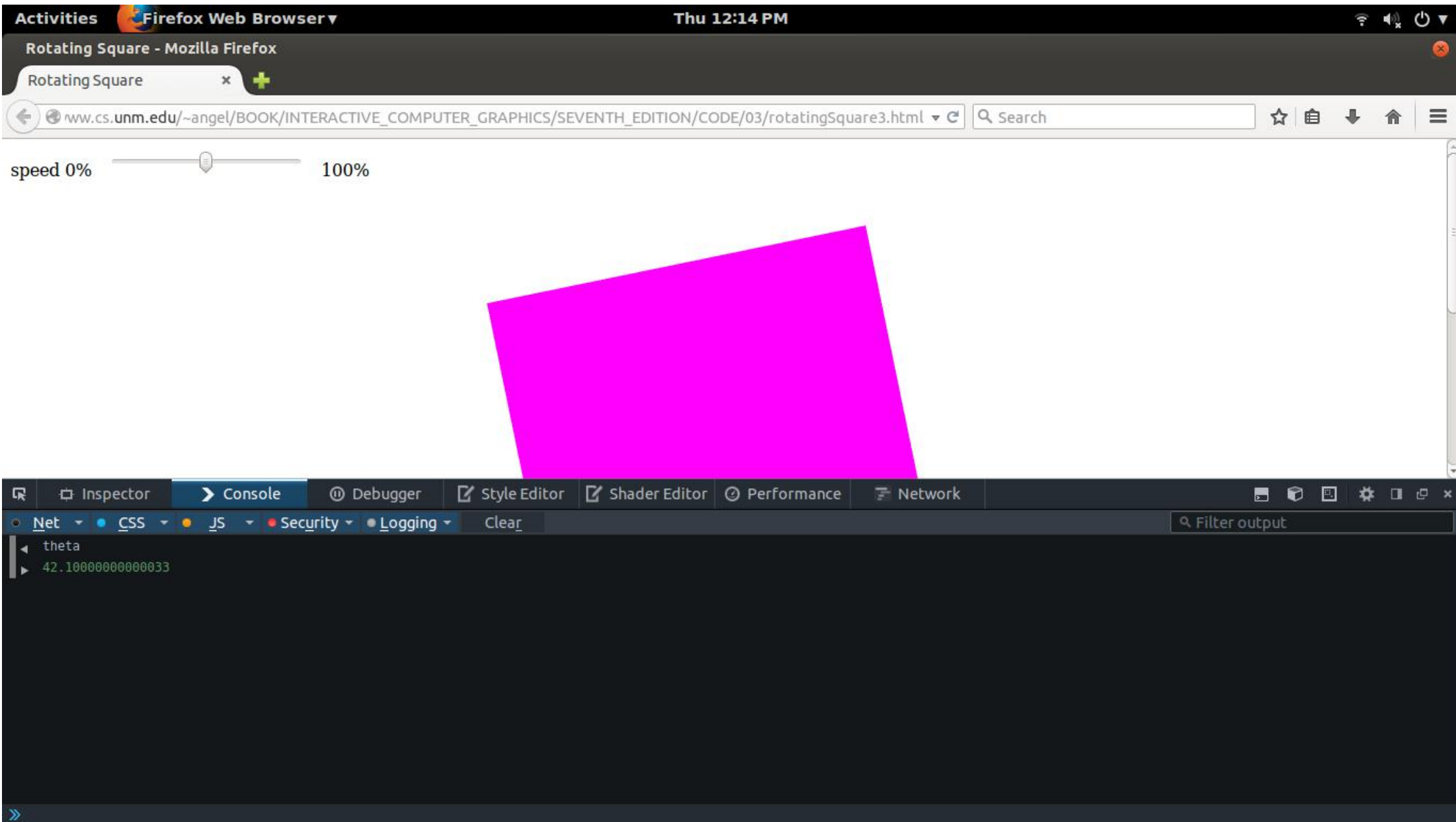
Program has resumed



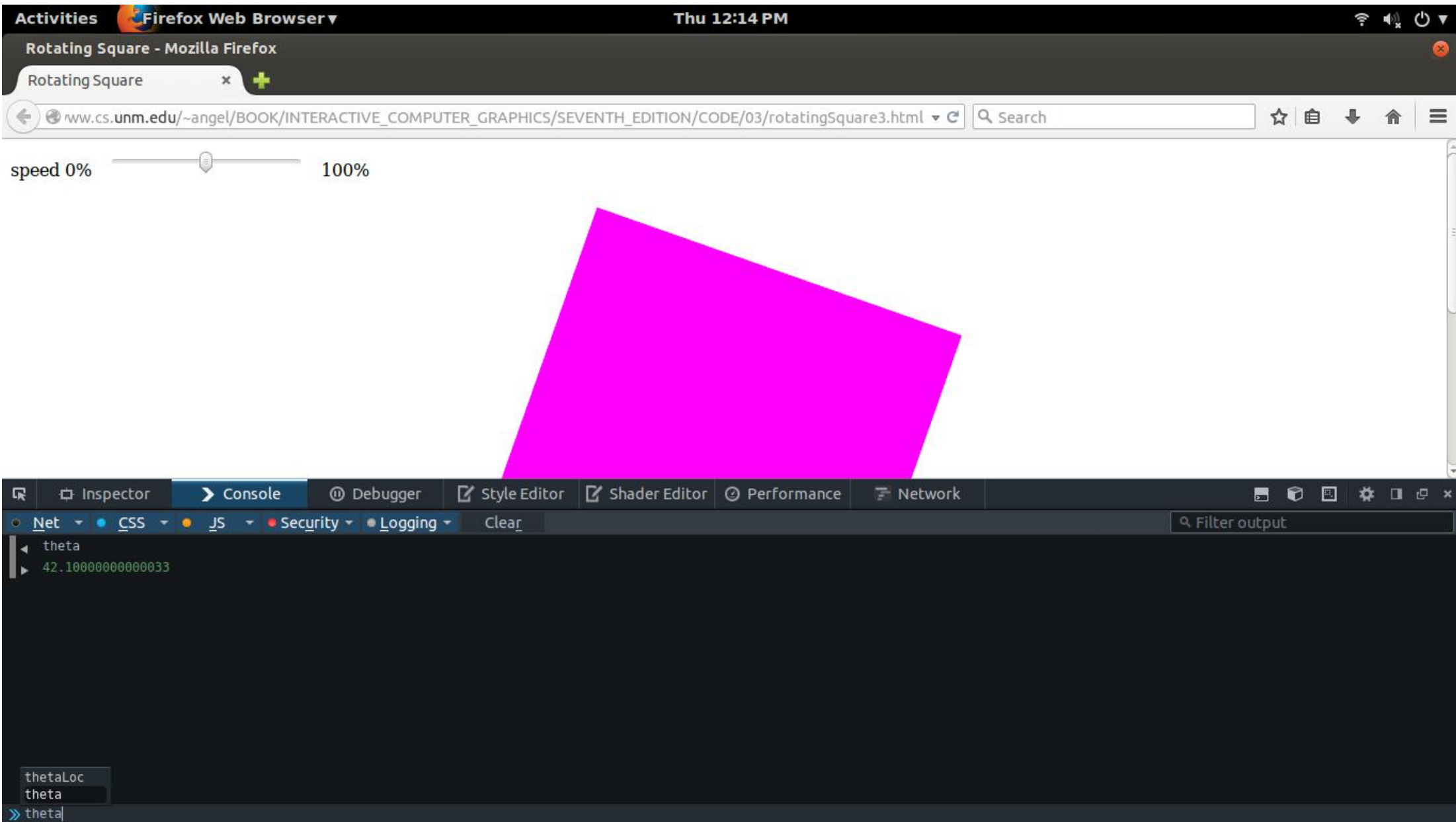
The console is a REPL style interpreter and contains the context of currently running scripts. Thus any variables or functions declared in the global scope of any script are accessible here.



Here I am accessing "theta" a variable declared in the global scope in the script "rotatingCube3.js"

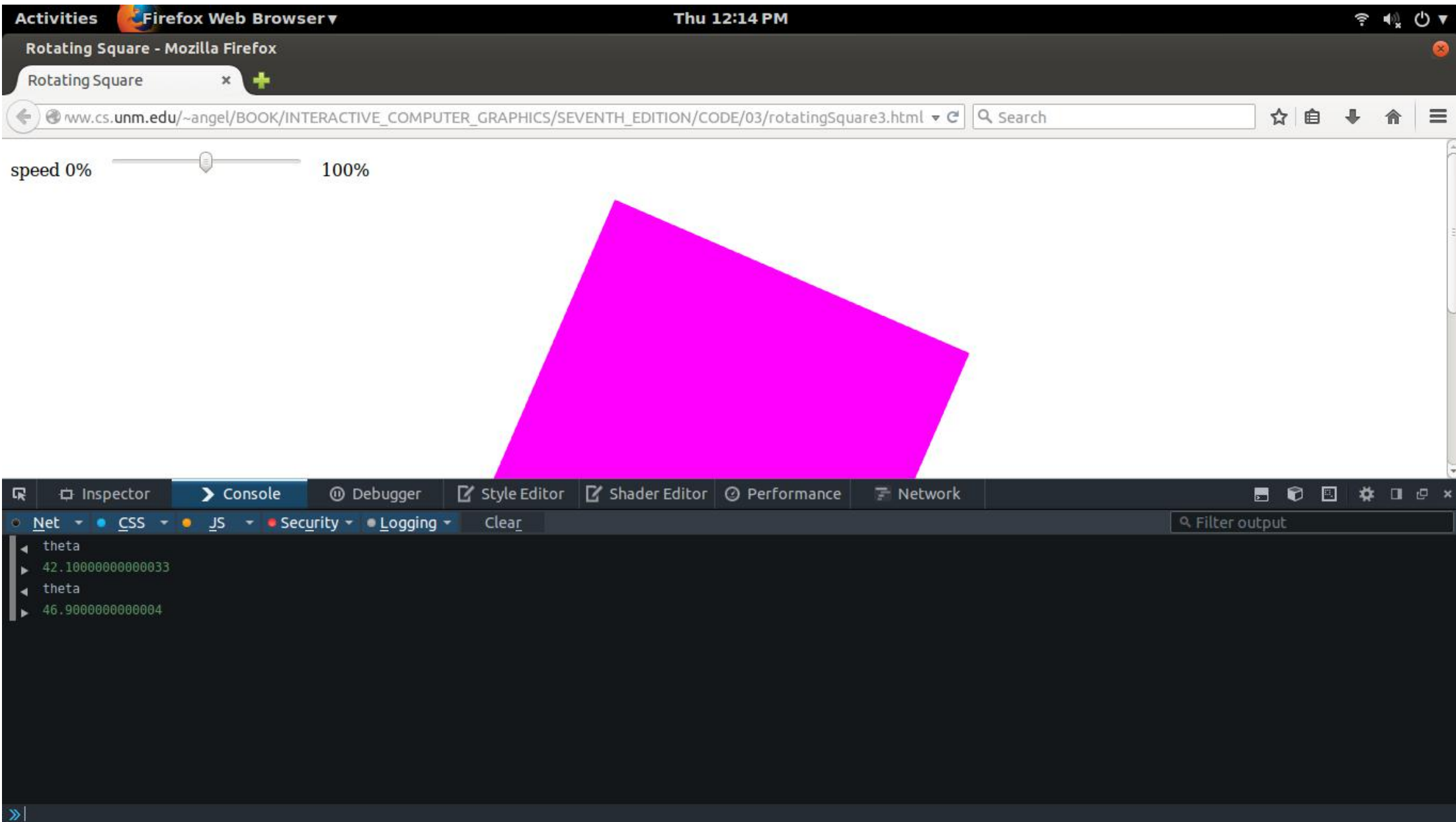


The console replies by echoing the value of theta.

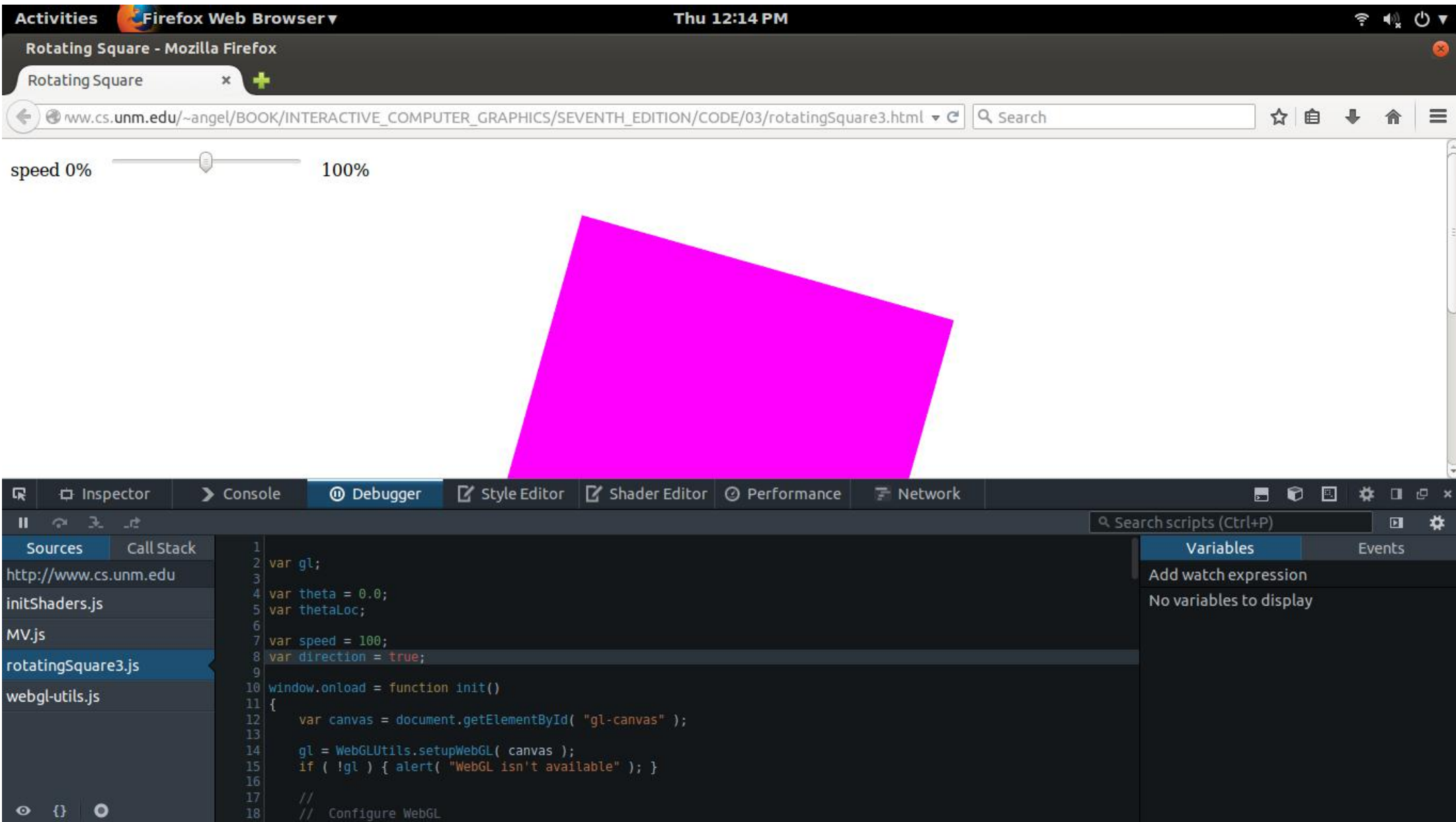


I enter the variable theta again.

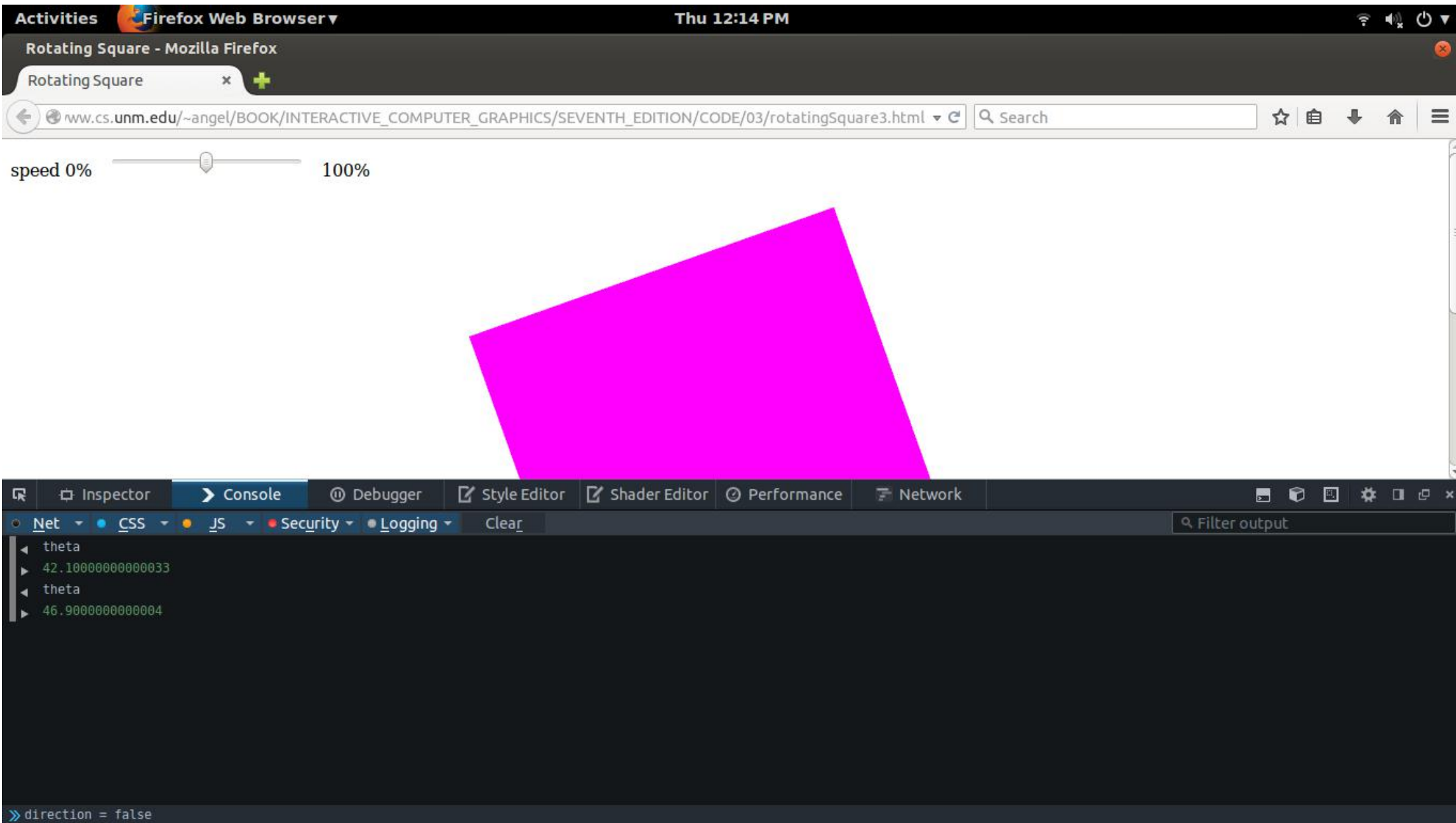




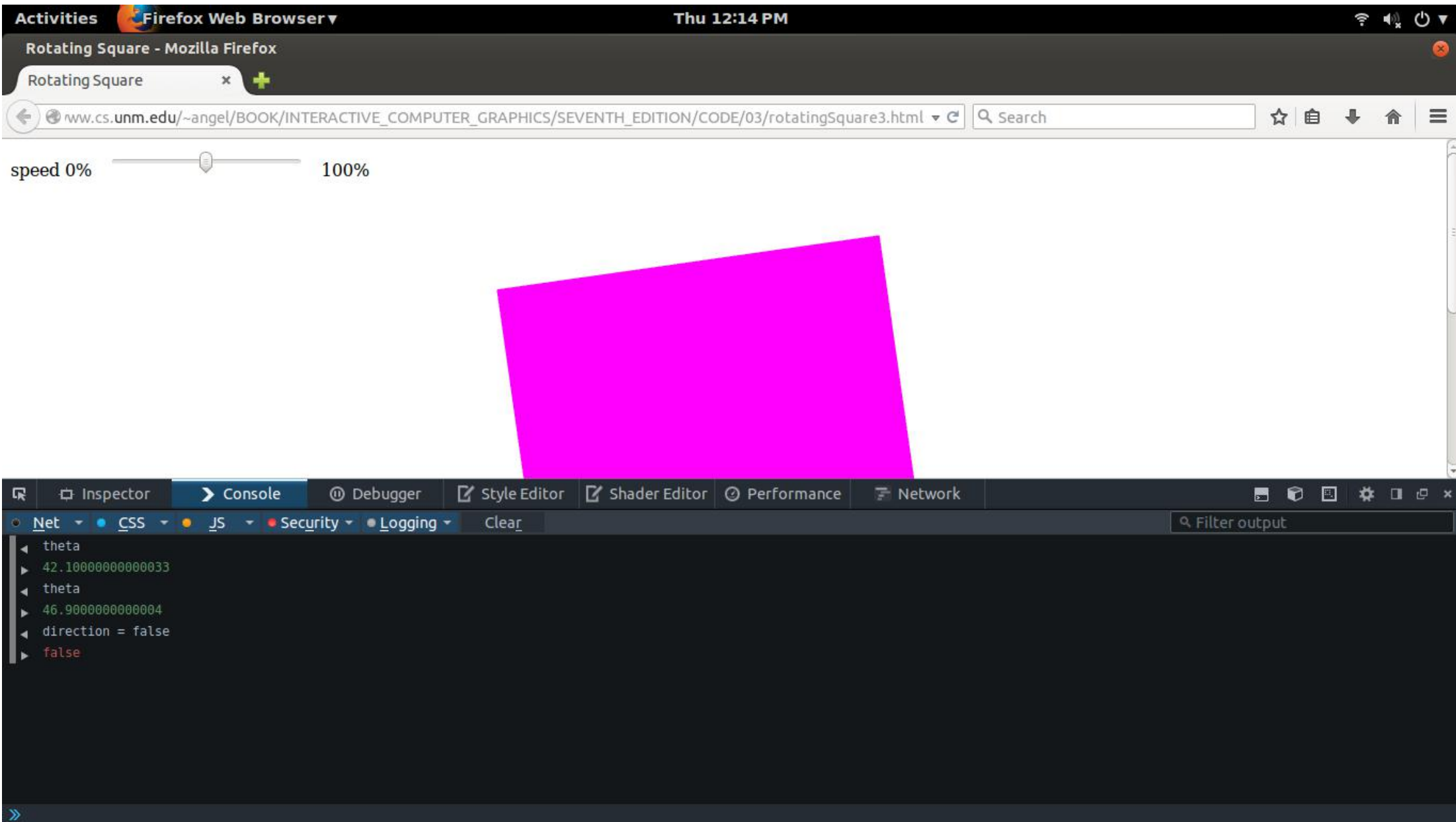
And the console replies by echoing the current value of theta. We can confirm that the value of theta is being modified in the loop in "rotatingSquare3.js" by reading these values.



Here is the variable `direction`, declared as a boolean type, in the global scope in "rotatingCube3.js"



I will now modify the value of direction to be "false", this should have the effect of changing the orientation of rotation of the cube from counterclockwise to clockwise around the Z axis.



These are the three main tools in developing and debugging WebGL programs in Firefox, and they have an abundance of functionality.