NMATH2 DEV LOG

**21 July 2014**

* Implemented successfully rotation object on SoftGraphView

**16 July 2014**

* TODO: if we want to rotate the object along touching, we have to anchor coordinator, so we must to implement another coordinator and rotate it, other objects will be drawn on the second coordinate.

This

gl\_Position = projection \* model \* view \* vec4(vertexPosition, 1);

is **wrong**. Matrix multiplication is not commutative, i.e. the order of operations matters. The transformations on a vertex' position, in order are:

* model
* view
* projection

Matrix multiplication for column vectors as used by OpenGL is left associative, i.e. goes from right to left. Hence the expression in the R side of the statement **should be**

gl\_Position = projection \* view \* model \* vec4(vertexPosition, 1);

However you can contract view and model transform into a compound modelview (first model, then view) transform. This saves a full matrix multiplication

gl\_Position = projection \* modelview \* vec4(vertexPosition, 1);

The projection should be kept separate as other shading steps may require the eye space position of the vertex which is the result of modelview \* position without projection applied.

BTW: You're transforming the vertex position, not the data. A vertex consists a larger number of attributes (not just the position) hence calling it "Data" is semantically wrong.

What is the best way to translate and rotate objects?

Those are part of the modelview transform. You should create a transformation matrix exactly *one* time on the CPU and pass it to the GPU. Doing this in the shader would force the GPU to redo the whole calculation for each and every vertex. You don't want to do this.

<http://stackoverflow.com/questions/14084071/opengl-rotation-and-translation-done-correctly>

**15 July 2014**

FunctionPlotter: I found that I had expressed matrices in row-major but I used them with Android Matrix class which consider that matrices are in column-major.

**14 July 2014**

* Draw OK with opengl using GL\_LINES

**13 July 2014**

* Draw the graph with opengl, but it was not good.

**04 July 2014**

* Criteria

**03 July 2014**

* Criteria
* Change interval tree in two case “var elemen\_of” and number < var < number

**02 July 2014**

* Fix bug in parseFunct
* Test with f(x)=sin(x OK

**01 July 2014**

* Re-implement domain method in the way just like paseFunct.
* Test f(x,y)=sin(x)+cos(y) OK with derivative

**30 June 2014**

* Install libglut into Cygwin in order to build freeglut-2.8.1
* Fix bug count gParenTop in functionCall (Linux version)
* Get back to nlabparser.c
* Test f(x,y)=sin(x)+cos(y) failed with Linux version
* Updated nlabparser.c, now it parse ok above function

**29 June 2014**

* Fix some bugs from nlabparser2 of linux.
* Updated makefile of linux.

**15 December 2014**

* Implement virtual *Criteria\* andSelf(Criteria& c)* method at Criteria and overload it at SimpleCriteria and CompositeCriteria.