forGL Information

GitHub has best information

https://github.com/RMax2015/forGL

Haxe Community Forum
search: 4GL or forGL
https://community.haxe.org

forGL is not a programming language but may act like one

forGL Goals

Anyone that can type on a computer* can learn computer programming**.

Computer programming experience or even English*** is not needed.

* Such as: Phones, Tablets, Laptops, Consoles, Desktops…

** Languages: Haxe, Haxe output languages, forGL itself...

*** Concept proved. More Internationalization work to do!

forGL is not a programming language but may act like one

forGL Inputs

Natural Languages

Must be text based and simplified (European now)

Dictionary approach as common way

Mathematics as a Language

Simplified : (now) No Symbolic Math

Future: 2D & 3D plot rendering

forGL NL Simplification

Natural Languages (NL) simplified to:
Verbs, Nouns, Operators, Punctuation.

Key Idea !
NL Dictionaries are widely known/used.
forGL Dictionary words <==> programs.

forGL Dictionary definitions :
easily read and changed by User
and run by forGL Interpreter.

ForGL NL Simplification

Not really Natural Language processing

Lexing: no recursion, 1 pass
Parsing: no recursion, 1 resolve* pass
Other passes reduce Interpreter work
or support features: Export As Code...

* Resolve pass: Infer obvious data.
Token search in Dictionary & any not
found set as Unknown and ignored.
Inferred as variables IF referenced.

forGL NL Complexity

NL word order is Complex :
English mostly uses Subject Verb Object.
I (s) like (v) Haxe (o).

https://en.wikipedia.org/wiki/Word_order
Gives other 5 word orders
and various exceptions
on a per Natural Language basis.

Key Idea ! forGL will not force word order except as selected by User: such as to support Export As Code.

forGL NL Complexity

NL Parts of Speech :

English has between 10 to 20? parts of speech.
Greek has THOUSANDS.
forGL simplifies to: Verbs, Nouns,
Operators (from Math) and Punctuation.

NL Reading Order :

most Left to Right.
Hebrew & Arabic & others Right to Left.
Classical Chinese Top to Bottom
with Leftmost column first.

forGL NL Complexity

NL Punctuation :
I think some written languages do not even use punctuation characters.

Instead I think just gaps represent punctuation. (Chinese? Others?)

Not sure.
I am no Linguist.

forGL and Math

Math Notation : depends on context. Different branches of Math use same notation to mean different things. Same notation may be used with different order of appearance.
Infix is typical & some prefix and postfix.

Math Concepts :
Absolute accuracy and exact precision may not fit well with computer Hardware or programming Software.

forGL and Math

Math Concepts :
Pure Math requires large effort to do correctly (consider Haskell).

forGL uses an Engineering approach.
Allows acceptable results for (hopefully) a wide spectrum of use cases.
1 to 1 mapping of forGL Math words in Dictionary to Haxe Math support.

forGL as an Application

Enable Beginning Programmers :

Simple things done in a simple way.
Full control of forGL options.
Options have usable defaults.
Display intermediate results as wanted.
Intermediate results help with
    Understanding AND Debugging.
Informative Warnings and Errors.
Use Color to show: Nouns, Verbs, etc.
Overall ... use Best Known Practices.

forGL as an Application

Enable Programmer Learning :

forGL allows Experimentation with :

Paradigms :

Imperative vs Declarative (Future goal)
Procedural vs Concatenative.
Concurrency & Objects added soon?

Notations :

Prefix, Infix, Postfix allowed.

Multiple ways to express Problems.
Use what seems best to you.

forGL as an Application

Enable More Programmer Learning :

forGL Inference approach allows :
Use of text with various words not in Dictionary. Words not found are treated as simple passive Local Noun names and are ignored until used.
No Warnings or Errors for unknowns.

Dictionary approach allows use of a mix of real or fictional language words.

forGL as an Application

Enable Programmer Productivity :

Key Ideas: Export As Code, Comments, Export As Application (soon?)

forGL Verb selected to Export As Code.
forGL syntax changed to Procedural.
Procedural syntax is run by Interpreter.
Avoids trying to Export some bugs.
Comments are Exported as well to allow Semantic Traceability.

forGL as an Application

Export As Code :

Key Ideas ! No other 4GL app allows Export As Code. No other programming language environment Exports to Haxe.

A HA ! ! ! Moment :

Haxe chosen as easy way to be dangerous with JavaScript. Using Göedel's Theorem & Thinking by Contract leads to ALSO using Haxe as output of forGL. A HUGE WIN ...

forGL as an Application

Dictionary As Application (soon?) :

Key Ideas: Entire Dictionary and not just 1 Verb Exported As Code. CLI args and Menu added to support Automated running or Manual running. Args / Menu items refer to Verbs in Dictionary.

Using Göedel's Theorem & Thinking by Contract leads to ALSO extending Export As Code.

forGL Testing Needed !

Test CLI on Windows, Linux and Mac
Test European languages
Test other Left to Right languages
(after i18n) Translation checking:
    Test Messages, UI Labels...
(after JavaScript fix) Test in Browser
Test Languages with no Punctuation
(after GUI added) Test GUI features

(later) Test Right to Left languages

forGL Changes Soon?

Add Automated Testing support (.toml)
Fix JavaScript to allow Testing & use
Fix Export As Code, Add Dict As App
Start Internationalization (i18n)
Start support for Right to Left languages
Add to API for other apps use, Games...
Refactor forGL without UI as a Haxe Lib
Add HaxeUI for Windows, Linux, Mac
CLI to use simplified Curses approach
Add native Mobile, Console, ... with GUI

Amber Language (4GL)

Gerard W. Horgan, Willem van Rijk & others helped create/sell Amber with IDE. Proprietary ... 80's to early 2000's?

Amber did several European languages. Amber had Nouns, Verbs, Adjectives & Adverbs. Dictionaries had words in various languages and may switch as wanted. Had unlimited Math & newer Windows IDE. Apps in Amber could create GUI & buttons would run a Verb.

Forth Language

Charles H Moore created Forth 1970

Concatenative paradigm and others.
Dictionary of editable words to run.
Interpreter and Compiler.
Good fit for embedded or pre OS boot.

Https://en.wikipedia.org/wiki/Forth_(programming_language)