**Hydro Language Ruleset**

**Hydro Grammar Summary**

nonPoint – represent symbol rules in the grammar

EndPoint – represent symbol endpoint in grammar (Tokens from lexer)

( ) – Groups symbols together

| - Acts as an OR between symbols

\* - Indicates preceding symbol group can be optional or can occur any number of times;

+ - Indicates preceding symbol group has to occur at least once

? – indicates preceding symbol group is optional or can occur once

**Declarations**

Declaration -> VariableDecl

| FunctionDecl

| ClassDecl

| statement

VariableDecl -> TYPE IDENTIFIER (EQUAL expression)? terminator

FunctionDecl -> FUNCTION method END FUNCTION

ClassDecl -> SETCLASS IDENTIFIER (INHERIT IDENTIFIER)? EOL

(Method declare) \*

CLASSEND

**Expressions**

Expression -> assignment

Assignment -> coal

logicalOr -> logicalXor (OR logicalXor) \*

logicalXor -> logicalAnd (XOR logicalAnd) \*

logicalAnd -> equality (AND equality) \*

statement -> block

| breakStmt  
| doStmt  
| exitStmt   
| ifStmt  
| forStmt  
| printStmt  
| quitStmt  
| returnStmt  
| selectStmt  
| whileStmt  
| expressionStmt

Helpful Rules

terminator → EOL | EOF

parameters → IDENTIFIER (COMMA IDENTIFIER)\*

arguments → expression (COMMA expression)\*

constructorDecl → CONSTRUCTOR LPAREN parameters? RPAREN EOL  
 block END CONSTRUCTOR EOL

methodDecl → METHOD method END METHOD EOL

foreignMethodDecl → FOREIGN METHOD method END METHOD EOL

staticMethodDecl → STATIC METHOD method END METHOD EOL

method → IDENTIFIER LPAREN parameters? RPAREN EOL block

getter → GET IDENTIFIER EOL block END GETTER EOL

setter → SET IDENTIFIER LPAREN IDENTIFIER RPAREN EOL BLOCK EOL

range → expression (TO | DOWNTO) expression

then → THEN (breakStmt | continueStmt | exitStmt | returnStmt | expressionStmt) terminator