**­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­TWT PROGRAMMING LANGUAGES**

CS 150 Machine Problem

**User Manual**

Gueco, Maria Rosario

Juntado, Marbille

Lenon, Mikaela Jun

Navarro, Gabriel Kelly

Arceo, Jahziel Rae

CS 150 HTWX

­­­­­­­INTRODUCTION

* 1. **Language Name:** TWT Language

Formal/Official/Long name: The Wanna Tweet Language

The programming language we created is inspired from the famous social networking website Twitter. The syntax, the rules, and terms mostly came from the said application. The programming paradigm of this language is Imperative for it is straight forward and just simply follows the expression given line by line.

* 1. **Grammar Definition**

<Program> -> <Declaration> <Main>

| <Main>

<Declaration> -> <Dtype> <Vname> "(" <Args> ")" "{" <Block> <Return>"}"

| <Dtype> <Vname> "(" <Args> ")" "{" <Block> "}"

<Main> -> "LOGIN" <Block> "LOGOUT"

| "LOGIN" "LOGOUT"

<Block> -> <State> // Block is made up of statement <State> of the same level

<State> -> <Loop> "#" <StatePrime>

| <If> "#" <StatePrime>

| <Assignment> "#" <StatePrime>

| <Call> "#" <StatePrime>

| <Printing> "#" <StatePrime>

| <Control> "#" <StatePrime>

<StatePrime> -> <Loop> "#" <StatePrime>

| <If> "#" <StatePrime>

| <Assignment> "#" <StatePrime>

| <Call> "#" <StatePrime>

| <Printing> "#" <StatePrime>

| <Read> "#" <StatePrime>

| <Control> "#" <StatePrime>

<Loop> -> "RT" <Boolean> {" <Block> "}"

<If> -> "IF" <Logic> "FOLLOW" "{" <Block> "}" <ElseIf> <Else>

| "IF" <Logic> "FOLLOW" "{" <Block> "}" <Else>

| "IF" <Logic> "FOLLOW" "{" <Block> "}"

<ElseIf> -> "ELSEIF" <Logic> "FOLLOW" "{" <Block> "}" <Elseif>

<Else> -> "ELSE" <Logic> "FOLLOW" "{" <Block> "}"

<Assignment> -> <DType> <Vname> "=" <Vname>

|<DType> <Vname> "=" <Reading>

|"@INT" <Vname> "=" <INT>

| "@CHIRP" <Vname> "=" <CHAR>

| "@COKE" <Vname> "=" <FLOAT>

| "@MSG" <Vname> "=" <STRING>

| "@TRALSE" <Vname> "=" <BOOL>

| <DType> <Vname> "=" "(" <Exp> ")"

<Call> -> <Vname> "(" <Args> ")"

<Printing> -> "TWEET" <Vname>

| "TWEET" <EXP>

| "TWEET" <INT>

| "TWEET" <FLOAT>

| "TWEET" <CHAR>

| "TWEET" <STRING>

| "TWEET" <BOOL>

<Reading> -> "REPLY" <Vname>

| "REPLY" <EXP>

| "REPLY" <ID>

<Control> -> "UNFOLLOW" | "LIKE" | "BLOCK"

//This part is the one we made sa ME compy na lng natin

<Exp> -> <Term> <ExpPrime>

<ExpPrime> -> "+" <Term><EspPrime>

| "-" <Term><ExpPrime>

<Term> -> <Fact><TermPrime>

<TermPrime> -> "\*" <Fact><TermPrime>

| "/" <Fact><TermPrime>

<Fact> -> <INT> | <CHAR> | <FLOAT> | "(" <Exp> ")"

<Newline> -> "\n" <Newline>

<Logic> -> "(" <LogicCond> ")" | "~" <Logic>

<LogicCond> -> <Vname> <LogicOP> <Vname>

| <Vname | INT | CHAR | FLOAT > <LogicOP> <Logic>

| <Logic> <LogicOP> <Logic>

| <Logic> <LogicOP> <Vname | INT | CHAR | FLOAT >

| <Vname | INT | CHAR | FLOAT > <LogicOP> <Vname | INT | CHAR | FLOAT >

<LogicOP> -> ">=" | "<=" | "==" | ">" | "<"

<Dtype> -> "@INT" | "@CHIRP" | "@COKE" | "@MSG" | "@TRALSE"

<Args> -> <Dtype> <Vname> "," <Args>

<Newline> -> "\n"<Newline> | "\n"

<Return> -> "REPORT" <ID> | "REPORT" <Vname> | "REPORT" <Exp>

* 1. **Lexical and Syntax Analysis**

List of Lexemes and their tokens:

t\_dec = 3 # @INT

float\_dec = 4 # @COKE

char\_dec = 5 # @CHIRP

string\_dec = 6 # @MSG

bool\_dec = 7 # @TRALSE

login = 1 # @LOGIN

logout = 0 # @LOGOUT

if\_state = 20 # IF

elseif\_state = 21 # ELSEIF

else\_state = 22 # ELSE

loop\_state = 30 # RT

break\_state = 31 # UNFOLLOW

continue\_state = 32 # LIKE

exit\_state = 33 # BLOCK

exec\_state = 34 # FOLLOW

read\_state = 10 # REPLY

print\_state = 11 # TWEET

return\_state = 12 # REPORT

TRUE = 13 # YES

FALSE = 14 # NO

NOT = 15 # '~'

INT = 71

FLOAT = 72

CHAR = 73

STRING = 74

openBrace = 41 # '{'

closeBrace = 42 # '}'

plusSign = 43

minusSign = 44

divSign = 45 # '/'

mulSign = 46

asSign = 47 # =

openParen = 48 # '('

closeParen = 49 # ')'

hashSymbol = 50 # '#'

commaSign = 51 # ',''

lessEqSign = 52 # '<='

greatEqSign = 53 # '>=''

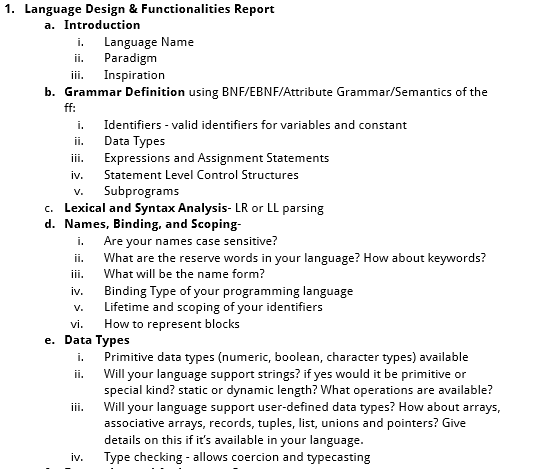
eqSign = 54 # '=='

lesserSign = 55 # '<'

greaterSign = 56 # '>''

VARIABLE = 19

nextString = " "



* 1. Names, Binding, and Scoping

Naming variables and writing Reserved words are both case sensitive. For the reserved words to be distinct, we’ve decided that they must be all capitalized. As for the variable, any combination of letters will do as long as it is not inside a quotation mark (“ “), it is not a string and hence a variable name. keywords?

* 1. Data Types

