Phrase Structure

Program ::= IDENT { Dec* Stmt* }
Dec ::= Type IDENT ;
Type ::= image | pixel | int | boolean
Stmt ::= ; | AssignStmt | PauseStmt | IterationStmt | AlternativeStmt
AssignStmt ::= IDENT = Expr ;
| IDENT = Pixel ;
| IDENT = STRING_LIT ;
| IDENT . pixels [Expr , Expr] = Pixel ;
| IDENT . pixels [Expr , Expr] (red | green | blue) = Expr ;
| IDENT . shape = [Expr , Expr] ;
| IDENT . location = [Expr , Expr] ;
| IDENT . visible = Expr ;
Pixel ::= {{ Expr , Expr , Expr }}
Expr :: = OrExpr (ϵ | ? Expr : Expr)
OrExpr ::= AndExpr (| AndExpr)*
AndExpr ::= EqualityExpr (& EqualityExpr)*
EqualityExpr ::= RelExpr ((== | !=) RelExpr) *
RelExpr ::= ShiftExpr ((< | > | ≤ | ≥) ShiftExpr) *
ShiftExpr ::= AddExpr ((≪ | ≫) AddExpr)*
AddExpr ::= MultExpr ((+ | -) MultExpr) *
MultExpr ::= PrimaryExpr ((* | / | %) PrimaryExpr)*
PrimaryExpr ::= IDENT
| INT_LIT
| BOOLEAN_LIT
| x
| y
| Z
| SCREEN_SIZE
| (Expr)
| IDENT [Expr , Expr] (red | green | blue)
| IDENT . height
| IDENT . width
| IDENT . x_loc
| IDENT . y_loc
PauseStmt ::= pause Expr ;
IterationStmt ::= while (Expr) { Stmt* }
AlternativeStmt ::= if (Expr) { Stmt *} | if (Expr) { Stmt* } else { Stmt* }