```
<program>            ==> <globalVars> <otherFunctions> <mainFunction>
<mainFunction>       ==> <stmts> | RBRACE
<stmts>              ==> <stmt><stmts> | RBRACE
<stmt>               ==> <gen_stmt> | <condnal_stmt>
<loop_stmts>         ==> <loop_stmt><loop_stmts>  | RBRACE
<loop_stmt>          ==> <lp_stmt> | BREAK SEMI_CL | CONTINUE SEMI_CL
<lp_stmt>            ==> <gen_stmt> | <loop_condnal_stmt>
<global_vars>        ==> <global_assignment><global_vars> | eps
<global_assignment>  ==> <primitive_type> ID ASSIGN_OP <const_value> SEMI_CL

<gen_stmt>           ==> ID  <func_ass_decl> | <other_stmt>
<other_stmt>         ==> <io_stmt> | <for_stmt>| <bot_stmt> | <comment>  | <unary_stmts>
                                 | <primitive_declaration_stmt> | <struct_def>

<func_ass_decl> ==>  ID <assign_more> | <array> <leftHandSide1> | <funcCall_stmt>

<comment>            ==> COMMENT_ST <ascii_text>
<ascii_text>         ==> ASCII_CHAR <ascii_text> | COMMENT_END

<struct_def>         ==> STRUCT ID LBRACE <declaration_stmts>
<unary_stmts>        ==> <prefix_op> ID <unary_stmt_more>
<unary_stmt_more>    ==> COMMA <prefix_op> ID <unary_stmt_more> | SEMI_CL


<declaration_stmts>  ==> <declaration_stmt> <declaration_stmts> | RBRACE SEMI_CL
<declaration_stmt>   ==> <primitive_declaration_stmt> | ID ID <assign_more>
|<point_decl_stmt> | <velocity_decl_stmt>

<primitive_declaration_stmt>   ==> <primitive_type> ID <assign_more>

<assign_more>        ==> ASSIGN_OP <expression> <assign_comma>
                       | LSQUARE <NUM> RSQUARE <array2>
                      |<assign_comma>
<assign_comma>       ==> COMMA ID <assign_more> | SEMI_CL

<array2>             ==> ASSIGN_OP <curly_values> <assign_comma>
                       | LSQUARE <NUM> RSQUARE <assign_2d>
                       |<assign_comma>
<assign_2d>          ==> ASSIGN_OP <meta_curly>  <assign_comma> |  <assign_comma>
<curly_values>       ==> LBRACE <values> RBRACE
<meta_curly>         ==> LBRACE <curly_values> <more_curliness>
<more_curliness>     ==> COMMA <curly_values> <more_curliness> | RBRACE
```

```
<point_decl_stmt>    ==> POINT ID <pv_rest>
<pv_rest>            ==> SEMI_CL | COL_ASSIGN LBRACE NUM COMMA NUM RBRACE SEMI_CL
<velocity_decl_stmt> ==> VELOCITY ID <pv_rest>

<bot_decl_stmt>      ==> BOT ID <bot_rest>
<bot_rest>           ==> SEMI_CL | COL_ASSIGN LBRACE ID COMMA ID COMMA <idNum> COMMA
<idNum>  COMMA BOOL RBRACE SEMI_CL


<assign_operators>    ==> ASSIGN_OP | PL_EQ



<type>               ==> <primitive_type> | <bot_type> | ID
<type_list>          ==> <type> <type_more> | VOID ID
<type_more>          ==> COMMA <type> <type_more> | ID
<primitive_type>     ==> INT | FLOAT |  BOOLEAN
<bot_type>           ==> POINT | BOT | VELOCITY


<assignment_stmt>         ==> <var> <lefthandSide1>
<lefthandSide1>      ==> <assign_operators> <rightHandSide>
                       | COMMA <var> <lefthandSideMultiple>
                       | COL_ASSIGN LBRACE <idNum> COMMA <idNum> <bot_or_point>
<bot_or_point>      ==> RBRACE SEMI_CL | COMMA <idNum> COMMA <idNum> COMMA <bool> RBRACE
SEMI_CL
<idNum>              ==> ID| NUM
<lefthandSideMultiple> ==> COMMA <var> <lefthandSideMultiple> | ASSIGN_OP <rightHandSide>
<rightHandSide>      ==> <expression> <rightHandSideMultiple>
<rightHandSideMultiple> ==> COMMA <expression> <rightHandSideMultiple> | SEMI_CL


<var>                ==> ID <array>
<array>              ==> LSQUARE <array_arithm_expr> <brack_pair1> | DOT <var> | eps
<brack_pair1>        ==> RSQUARE <array_2d> | COLON <array_arithm_expr> RSQUARE
<array_2d>           ==> eps | LSQUARE <array_arithm_expr> RSQUARE


<values>             ==> <value> <values_more>
<values_more>        ==> COMMA <values> | eps
<value>              ==>  <const_value> | ID <value_rest>
<value_rest>         ==> <funcCall_inside_Value> | <array>
```

```
<funcCall_inside_Value> ==> LPARA <funcCall_inside_Value_rest>
<funcCall_inside_Value_rest>   ==>   <arg_list> | RPARA

<bool>               ==> TRUE | FALSE
<const_value>        ==> NUM | RNUM | <bool>

<loop_condnal_stmt>  ==> IF LPARA <expression> RPARA LBRACE <loop_stmts>
                           <loop_condnal_stmt_more>
<loop_condnal_stmt_more>  ==> ELSE <loop_condnal_stmt_even_more> | eps
<loop_condnal_stmt_even_more> ==> <loop_condnal_stmt> | LBRACE <loop_stmts>


<condnal_stmt>       ==>   IF LPARA <expression> RPARA LBRACE  <condnal_stmt_suffix>
<condnal_stmt_suffix> ==>  <stmts> <condnal_stmt_more>

<condnal_stmt_more>  ==> ELSE <condnal_stmt_even_more>| eps
<condnal_stmt_even_more> ==> <condnal_stmt> | LBRACE <stmts>



<expressions>        ==> <expression><more_expression>
<more_expression>    ==> COMMA <expression><more_expression> | eps


<array_arithm_expr>  ==> <arr_mul_div_expr><array_arithm_expr'>
<array_arithm_expr'> ==>  <sum_ops><arr_mul_div_expr><array_arithm_expr'> | eps
<arr_mul_div_expr>   ==> <arr_un_expr><arr_mul_div_expr'>
<arr_mul_div_expr'>  ==>  <mul_ops> <arr_un_expr><arr_mul_div_expr'> | eps
<arr_un_expr>        ==> <prefix_op><arr_end>
<arr_end>            ==> <idNum> | LPARA <array_arithm_expr> RPARA



<expression>         ==> <and_expr><expression'>
<expression'>        ==> LOG_OR <and_expr> <expression'> | eps
<and_expr>           ==> <reln_expr><and_expr'>
<and_expr'>          ==> LOG_AND <reln_expr><and_expr'> | eps
<reln_expr>          ==> <add_sub_expr><reln_expr'>
<reln_expr'>         ==> <rel_op> <add_sub_expr> <reln_expr'> | eps
<rel_op>             ==> LT | GT | LOG_EQ | LTE | RTE
<add_sub_expr>       ==> <mul_div_expr><add_sub_expr'>
<add_sub_expr'>      ==> <sum_ops> <mul_div_expr> <add_sub_expr'>| eps
```

```
<sum_ops>            ==> PLUS | MINUS
<mul_div_expr>       ==> <un_expr> <mul_div_expr'>
<mul_div_expr'>      ==> <mul_ops> <un_expr> <mul_div_expr'> | eps
<mul_ops>            ==> MULTIPLY | DIVIDE
<un_expr>            ==> <prefix_op><typecast> | <typecast>
<prefix_op>          ==> UNARY_INCR | UNARY_DECR
<typecast>           ==> <value> | LPARA <expression_more>
<expression_more>    ==> <expression> RPARA | <primitive_type> RPARA LPARA
                         <expression> RPARA



<arg_list>           ==>  ID <arg_list_more> | <const_value> <more42>
<arg_list_more>      ==>  ASSIGN_OP <value> <arg_list_more2> | <array_fn_vals> <more42>
<arg_list_more2>     ==>  COMMA <arg_list> | RPARA
<more42>             ==>  COMMA <id_array_const> <more42> | RPARA
<id_array_const>     ==>  ID <array_fn_vals> | <const_value>
<array_fn_vals>      ==> LSQUARE <array_arithm_expr> <brack_pair123> | DOT ID
<array_fn_vals> | eps

<brack_pair123>      ==> RSQUARE <array_2d0>
<array_2d0>          ==> eps | LSQUARE <array_arithm_expr> RSQUARE


<funcCall_stmt>      ==>  LPARA <funcCall_stmt_rest>
<funcCall_stmt_rest> ==>   <arg_list> SEMI_CL | RPARA SEMI_CL

<otherFunctions>     ==>  <function><otherFunctions> | MAIN LPARA RPARA LBRACE
<function>           ==>  FUNCTION <type_list> LPARA <parameter_list> LBRACE
                           <fn_stmts>
<fn_stmts>           ==>  <stmt><fn_stmts> | <function><fn_stmts> | RETURN <expressions>
                          SEMI_CL RBRACE

<parameter_list>     ==>  <type> ID <parameter_list_more> | RPARA
<parameter_list_more>==>  COMMA <type> ID <parameter_list_more> | ASSIGN_OP <const_value>
                         <default_arg_list> | RPARA
<default_arg_list>   ==>  COMMA <type> ID ASSIGN_OP <const_value> <default_arg_list> |
RPARA

<io_stmt>            ==> READI INPOP ID SEMI_CL
```

```
<bot_stmt>           ==> ADDV <idDotComma> <idDotSCL>
                         | FW ID COMMA <expression> SEMI_CL
                         | RT ID SEMI_CL
                         | <point_decl_stmt> | <velocity_decl_stmt> | <bot_decl_stmt>

<idDotComma> ==> ID <idDotCommaMore>
<idDotCommaMore> ==> DOT ID <idDotCommaMore> | COMMA
<idDotSCL> ==> ID <idDotSCLMore>
<idDotSCLMore> ==> DOT ID <idDotSCLMore> | SEMI_CL


<for_stmt>           ==> FOR LPARA <assignment_stmt> <expression> SEMI_CL <update_stmt>
RPARA LBRACE <loop_stmts>

<update_stmt> ==>  <assignment_stmt> | <unary_stmts>
```