Deliverable II: Lex Generator

This segment of the project is in charge of generating tokens that simplify common human instructions, so they can be later processed and traduced for the CPU Simulator. To generate the tokens it must be considered the different ways a human can generate a command with the same outcome.

Logic

The robot is only capable of understand MOVE and TURN directions, therefore, an alalysis was made upon each instruction, its components and the diverse phrasings.

The following list shows the parts of a human instruction:

1. **Parts of a sentence**

    1. Noun: The robot is addressed in the sentence as a being or its name.
    2. Politeness: Every instruction must be polite, including a please.
    3. Structure: The instruction can be phrased as a command or a question.

2. **Instruction MOVE**

    1. Movement action: Key word that asks specifically for movement
    2. Quantity: The quantity of spaces thar the robot should move
    3. Type of quanity: Blocks
    4. Movement Direction: Forward

3. **Instruction TURN**

    1. Turn action: Key word that asks specifically for a turn
    2. Quantity: 90, 180, 270, 360
    3. Type of quantity: Degrees

From the results, the follwing tokens were generated.

Tokens

TOKEN	DEFINITION
ROBOT	Robot | robot
NOUN	you
REQUEST	Could
PLEASE	please
MOVE	move
TURN	turn
DIRECTION	ahead

TOKEN	**DEFINITION**
BLOCKS	block | blocks,
DEGREEW	degrees | degrees,
CONJUNCTION	and | then
DIRECTION	forward | left | right | top | bottom | backwards
DEGREES	90 | 180 | 270 | 360
AMOUNT	[0-9]+

Sentences

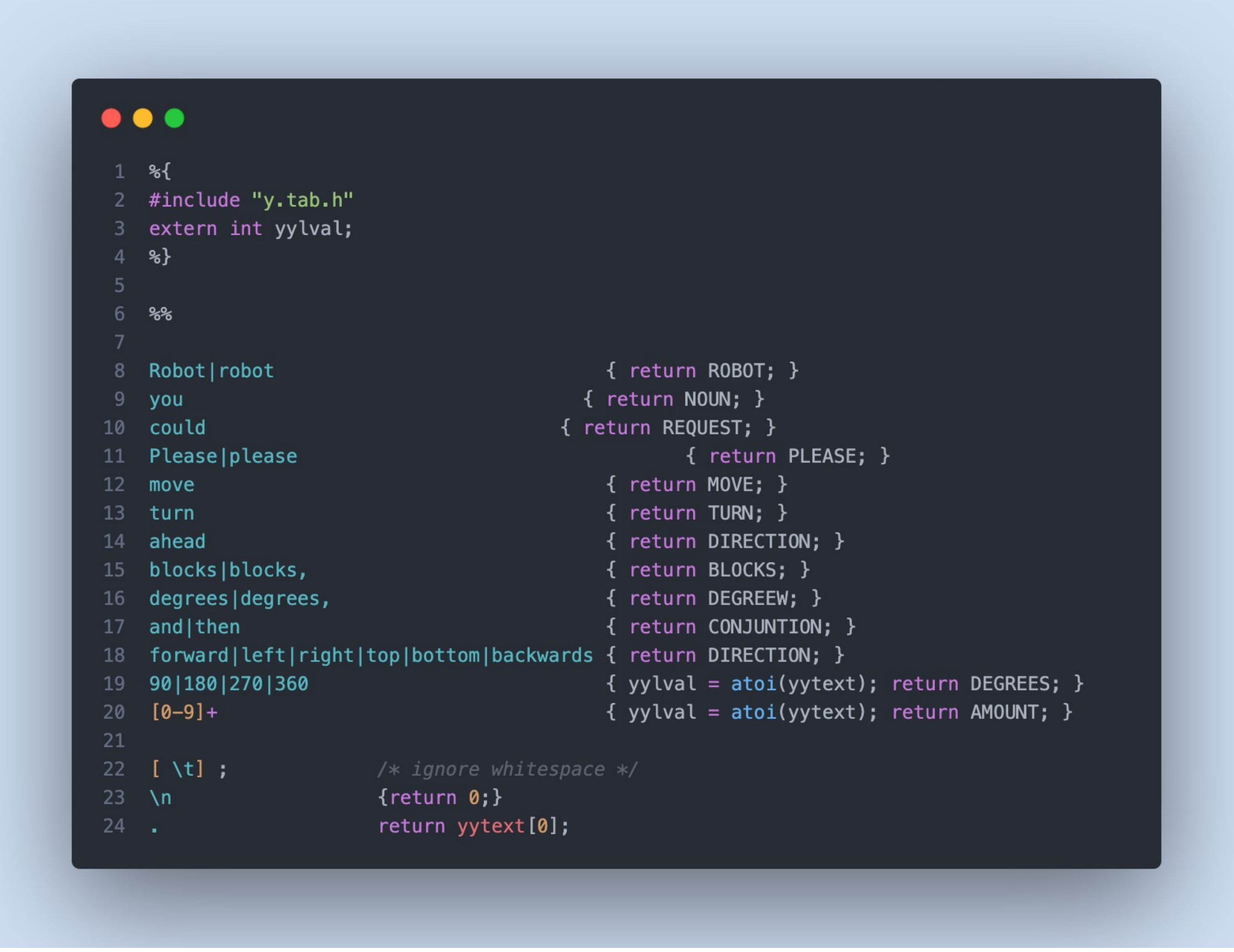
Valid Sentences

1. Robot please move 2 blocks ahead
2. Robot please move 2 blocks
3. Robot please move 3 blocks ahead and the turn 90 degrees, and then move 2 blocks
4. Robot please turn 270 degrees and then move 2 blocks
5. Robot could you please move 3 blocks?
6. Robot could you move 3 blocks please?
7. Robot move 3 blocks please
8. Please move 3 blocks

Invalid Sentences

1. Robot please move 2
2. Robot please turn 90
3. Robot move 3 blocks
4. Move 2 blocks and then turn 90 degrees
5. Robot move 5 ahead and turn 360 degrees
6. Robot please turn 20 degrees
7. Turn 90 degrees and then move 3 blocks
8. move 4 blocks

Code



```
%{
#include "y.tab.h"
extern int yylval;
%}

%%

Robot|robot                             { return ROBOT; }
you                                   { return NOUN; }
could                               { return REQUEST; }
Please|please                                  { return PLEASE; }
move                                    { return MOVE; }
turn                                    { return TURN; }
ahead                                   { return DIRECTION; }
blocks|blocks,                          { return BLOCKS; }
degrees|degrees,                        { return DEGREEW; }
and|then                                { return CONJUNTION; }
forward|left|right|top|bottom|backwards { return DIRECTION; }
90|180|270|360                          { yylval = atoi(yytext); return DEGREES; }
[0-9]+                                  { yylval = atoi(yytext); return AMOUNT; }

[ \t] ;              /* ignore whitespace */
\n                   {return 0;}
.                    return yytext[0];
```

Links

Github Main Project link

Github Deliverable II link