Files Included:

y.output
tetris.y
tetris.l
Makefile

Changes in lexer

- Collected directions and rotations into single tokens - DIR, ROT
- Added TICK while-loops
- Removed SPEED
- Added NOT
- Changed customisable shapes from 3x3 to 4x4

Syntax Directed Translation Scheme

States and Transitions are highlighted in y.output file. This file was made using yacc, which uses LALR(1) parser. As the number of states are huge we could not make a transition diagrams for the same

Explanation of Challenges

At present, none of the actions or symbols in the parser conflict with each other. We faced problems with the '=' symbol being used for initialization of the grid as well as for assignment in arithmetic operations. There were similar issues with the pause functionality which used the same symbol for the action as well as the key. These were both resolved by renaming the concerned conflict pairs.
The parser's shift-reduce conflicts are resolved by putting precedence for endifs and else statements, as they can result in ambiguities

Tests

1) Check variable definitions

```
Var_test init 32
test_var = Var_test
```

2) nested, if else

```
If 32>35
then one_more_thing + 24 //should give error as variable not initialized
```

```
            if not lost and Time < 10
                  5+3  //parser error
                  then  score + 2 Time +1 //testing multiple
            statements
            endif  //nested if
      else  score = 5
      Endif
```

3) Loops, grid calls

```
a) Do score = score + 2
   While not grid 5 3 //"grid num num" returns a bool

b) i init 0 //simple for loop implementation using do while loop
   Do i = i +1
   While i < 3

c) i init 0
   Do i = i +1
   While j init 0 j < 5 //should give error as "while stmt" is not
   in grammar
```

4) Conf checks: key press- to action sequence. Checks how keys are mapped to player inputs

```
Conf left KEY_T
Conf pause KEY_TAB
Conf clockwise KEY_RIGHTARROW
```

5) Grid initialization
   a) Does not let grid change anytime after initialzations

```
Grid = 32 32
```

6) Create new block

```
SHAPE_O False //removing O shape from generated blocks
Line1 True True True True //generating a custom 4x4 block
Line2 True False False False
Line3 True False False False
Line4 True False False False
```

7) Base running program

   Line clears, key mapping, and other base tetris things is handled by our compiler. Although these can be customized by the programmer too.

```
Count_x init 16
Count_y init 20
```

```
Grid init Count_x Count_y
Start init Count_x/2 Count_y

Count_ticks init 0
Tick
     Count_ticks = Count_ticks + 1
     If Count_ticks >=50
          then Time + 1
Tickend
```

Program toolchain

Our lex and yacc files generate a binary executable file which acts as a compiler, input can be directly given in terminal or taken from file. The compiler generates a c++ file: tetris.cpp which uses ncurses library for display management. The Makefile doing all this is included in the submissoin