Compiler Construction
Assignment - 1
Top Level Design Specs & Scanner Design

Group Details

Group Name - Tetreasy

Group Number - 39

Group Members :

Aman Sharma	2019A7PS0053G
Aarjav	2019A7PS0204G
Rudra Pratap Singh Chouhan	2019A7PS0164G
Shashank Gupta	2019A7PS0029G
Daksh Talreja	2019A7PS0014G
Ketan Raghunath Raut	2019A7PS0150G

Chapter 1 - Top Level Design Specifications

1. Overall Structure

Code written in our programming language will be scanned and parsed to create the target python file which can then be executed to play the game. Our programming language uses space to distinguish between tokens.

2. Offered primitives/Features

- Rows: to set the number of rows in the tetris grid.
- Columns: to set the number of columns in the tetris grid.
- Config: to set the keys corresponding to movement options.
- Speed: to set the speed of dropping pieces.
- timedgame: can be set to True or False depending on the requirement of timer in game.
- timer: to set the amount of time to be survived to win the game.
- music: to set music to on or off.
- nextq: to set the number of blocks visible in the next-queue.

3. Pipeline Schema

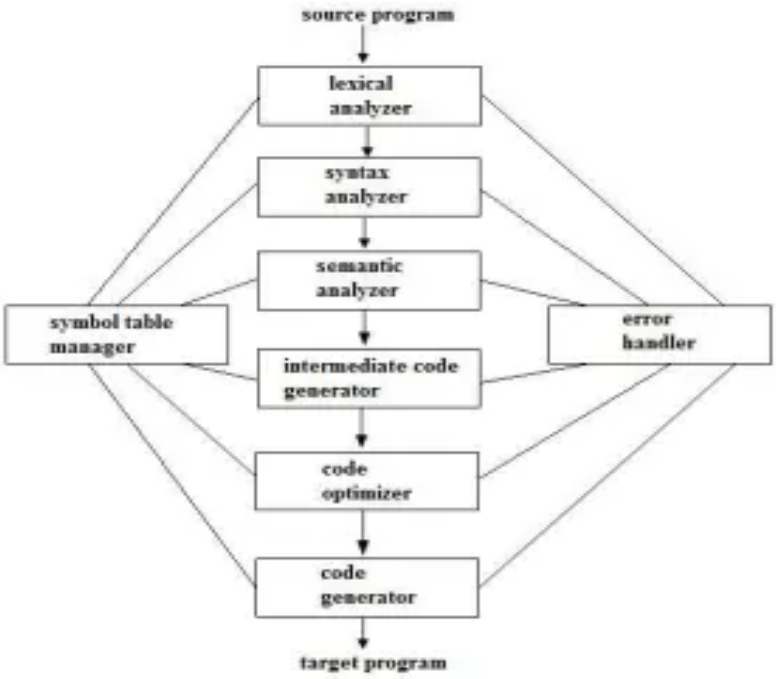

Fig 1.5 Phases of a compiler

- First, the code in the Tetris language will be sent to a scanner. Scanner will then return tokens of the input stream.
- Parser makes the Abstract Syntax Tree from the tokens provided.
- Semantic Analyzer checks for syntactical errors.
- After checking the program for syntax errors, an intermediate program would be generated from the code provided by the programmer in Tetris language.
- An executable is then made from this code which will finally be able to run Tetris.

Lexical Analyser/Scanner -The scanner is a subroutine which is frequently called by an application program like a compiler. The primary function of a scanner is to combine characters from the input stream into recognizable units called tokens. Overall, scanning is the process of reading the source code one character at a time in a methodical manner to convert them into tokens**.**

Syntax analyzer / Parser:
Parsing is the process of taking the tokens and generating a parse tree as the output. Parser is a program that does the parsing.

Semantic analyzer: semantic analysis is the process of drawing meaning from text. It allows computers to understand and interpret sentences, paragraphs, or whole documents, by analyzing their grammatical structure, and identifying relationships between individual words in a particular context

Target code generation: Target code generation is the last stage of compilation. Each line in optimized code may map to one or more lines in machine/assembly code.

Chapter 2 - Scanner Design

The table showcasing various lexemes, the token names and their attribute values.

Lexeme	Token Name	Attribute Value
=	assinop	AS
Any number	number	-
config	identifier	config
rows	identifier	rows
columns	identifier	columns
music	identifier	music
speed	identifier	speed
nextq	identifier	nextq
blocktype1	identifier	blocktype1
blocktype2	identifier	blocktype2
blocktype3	identifier	blocktype3
blocktype4	identifier	blocktype4
blocktype5	identifier	blocktype5
blocktype6	identifier	blocktype6
blocktype7	identifier	blocktype7
timer	identifier	timer
timedgame	identifier	timedgame
True	boolean	true

False	boolean	false
LEFT	move	left
RIGHT	move	right
SOFTDROP	move	softdrop
HARDDROP	move	harddrop
ROTATELEFT	move	rotateleft
ROTATERIGHT	move	rotateright

Chapter 3 - Division of Labor between the Scanner and Parser

We are using the scanner to identify errors in the tokens. The type of token is identified by the scanner. The parser creates the semantic trees.

Our scanner will take a sequence of characters (the source file of the program) and produce a sequence of tokens that will be used to feed the compiler's parser. It will consist of NEXTQ, LEFT, RIGHT, ROTATE(L/R), HARDDROP, SOFTDROP, etc. For example,

LEFT - It will shift the incoming block left.
RIGHT - It'll shift the incoming block right.
ROTATE - It will rotate the blocks.
HARDDROP - Force push down the incoming blocks vertically.
SOFTDROP - Push down the incoming blocks by 1 row.

Parser will interpret the component that breaks data into smaller elements for easy translation into another language. Our parser will do the syntax analysis, this process is called parsing. Furthermore, the parser checks whether the expression made by the tokens is syntactically correct or not.

Chapter 4- Division and Distribution of Roles and Responsibilities Among the team

As of now, the team has made combined efforts, with no specific division of tasks. All team members are aware of the specs and features of the designed system. We plan on dividing the work into tasks and assigning them to individuals as soon as the Mid semester exam ends. We also plan on keeping the roles flexible so that everyone can cover a wide variety of conceptual bases.