2020

Assignment - ASE

COMPONENT - 1

PRAVESH PANSARI

Contents

Simple Programming Environment

Interface:

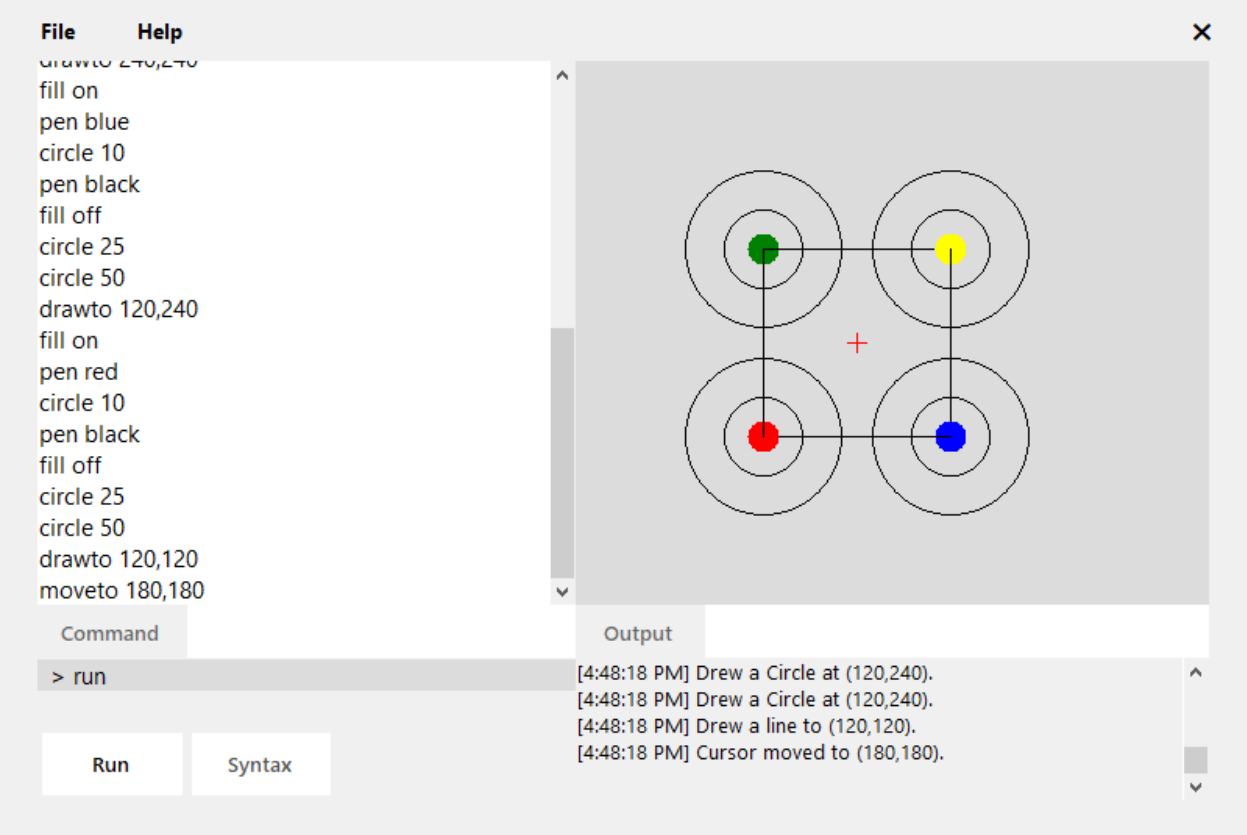


Figure 1: The GUI of Simple Programming Environment

This program creates a simple programming environment, where commands can be used to manipulate shapes and lines on an artboard. The artboard is the place where all drawings and the cursor are rendered. The menu allows access to different file operations, exit, and the about information. The current position of the cursor on the artboard is represented by the red cross. The command line runs single commands while the code editor can run multiple commands where every command is on a new line. Commands can be executed using the run button. The output box displays the log information and error information for the commands. The syntax button analyzes the code in the code editor and displays any errors it found in the log box.

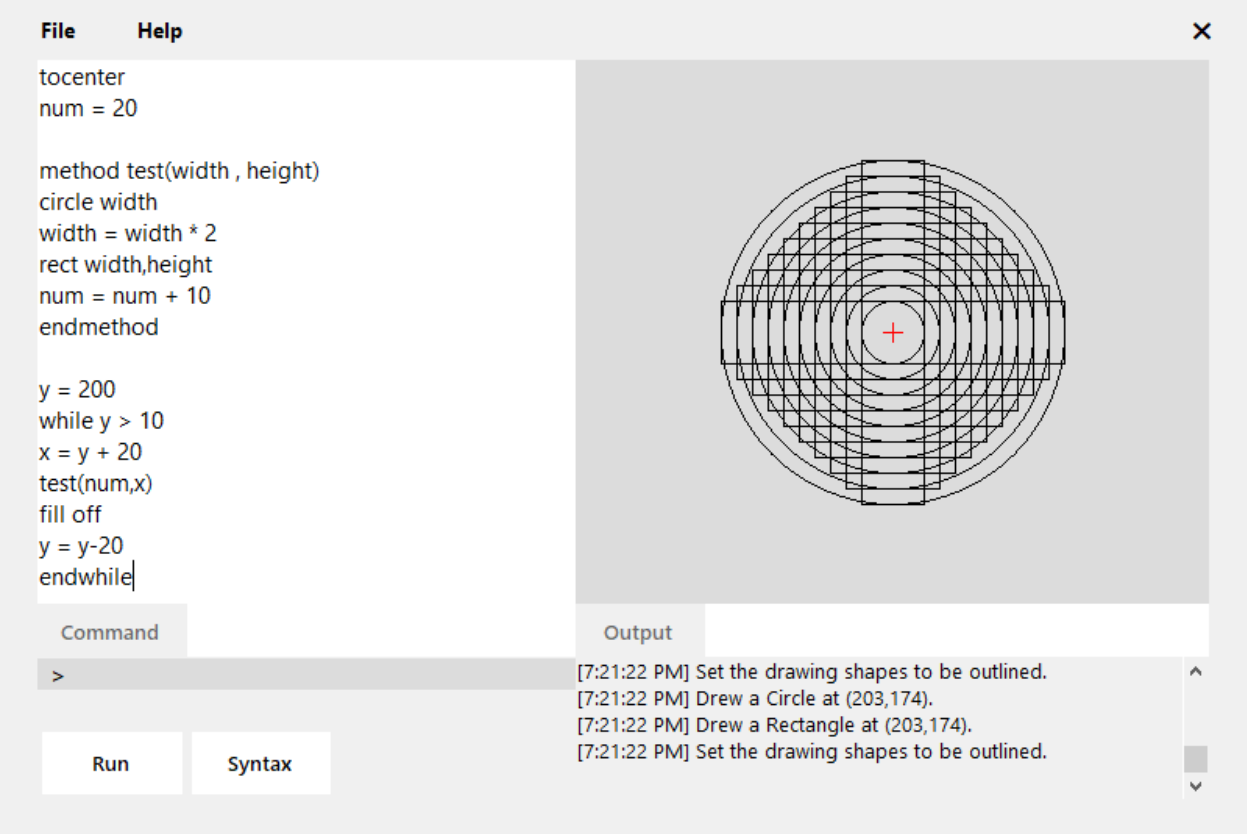


Figure 2: Making a complex drawing using advanced commands

Functions:

Four types of shapes can be drawn: rectangle, square, circle and triangle. And a line can be drawn given a destination position. The shapes can be drawn outlined or filled. The color of shapes and lines can be changed. The cursor can be reset to top left, moved to center or any position in the artboard. The board can be cleared, which removes all drawings on it. The file menu allows the user to create new file, save the code, open a file and exit the program. The help menu consists of information about the program.

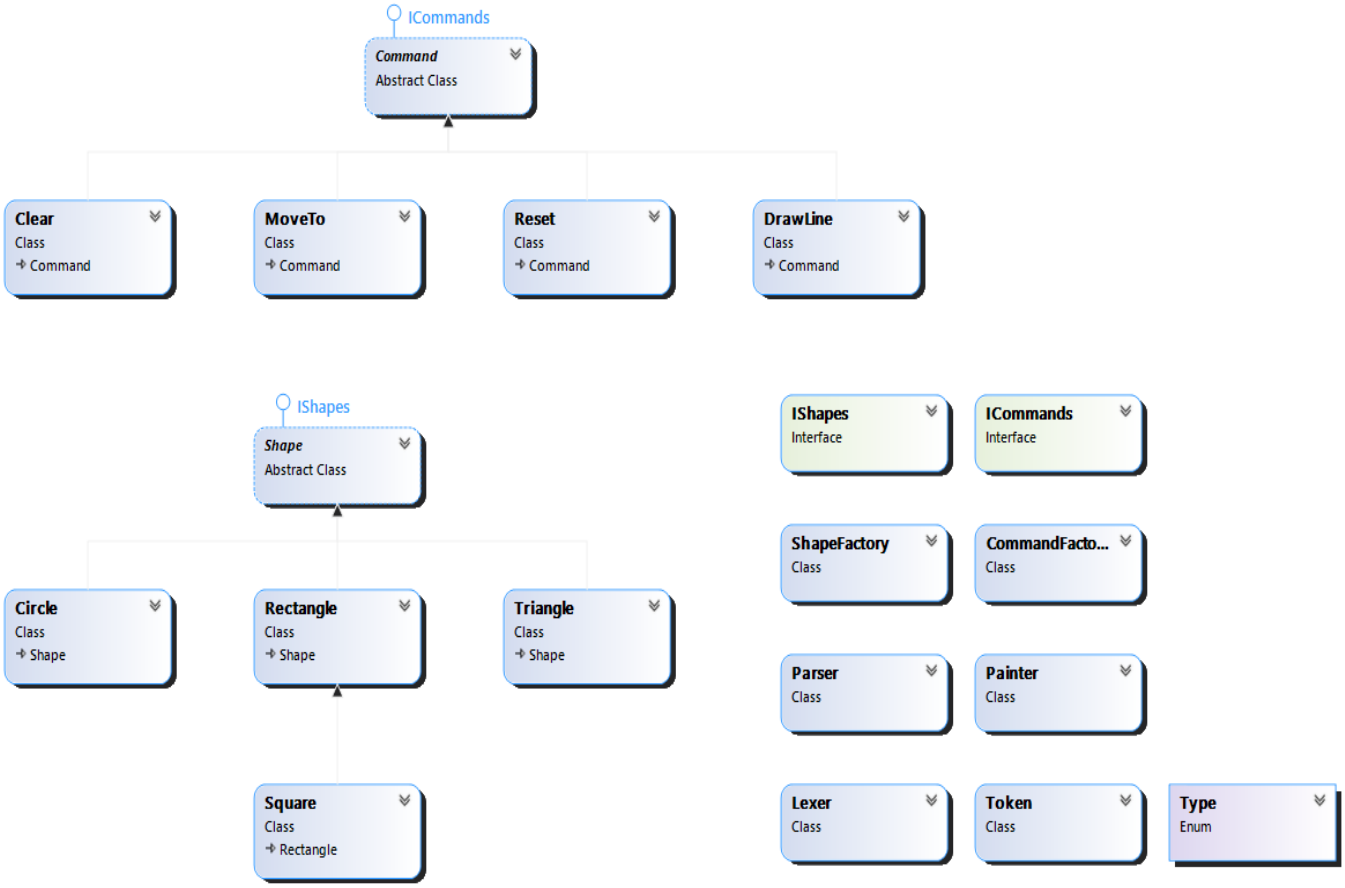
Variables can be declared in this program. They must contain only letters and can be assigned to integers only. Complex operations can be performed in the assignment statement including 5 types of mathematical operations. Commands can be run conditionally using if statements. They can be used as single line or a complete block of code. While loops can be used to redo commands until a stop condition. Methods can be declared and be called. Both parameterized and parameter less methods can be implemented. Then they can be called by passing appropriate parameters.

Commands:

- **drawto** int x, int y – Draws a line from current position to (x, y)
- **moveto** int x, int y – Moves the cursor to (x, y)
- **pen** string color – Changes the Pen color to color
- **fill** string input – Sets the fill of the shapes (on or off)
- **rect** int width, int height – Draws a rectangle of width and height
- **square** int size – Draws a square of length size
- **circle** int radius – Draws a circle of radius
- **triangle** int base, int height – Draws a triangle with base and height
- **clear** – Clears the drawing area
- **reset** – Reset the circle position to (0, 0)
- **tocenter** – Moves the cursor to center
- string test **=** int value – Creates a variable named test and assigns value to it
- **if** var **op** int x – Compares var against x using comparison operator op
- **while** var **op** int x – Loops till the condition var against x is true using op
- **method** example (param) – Creates a function example with parameter param

Program Design

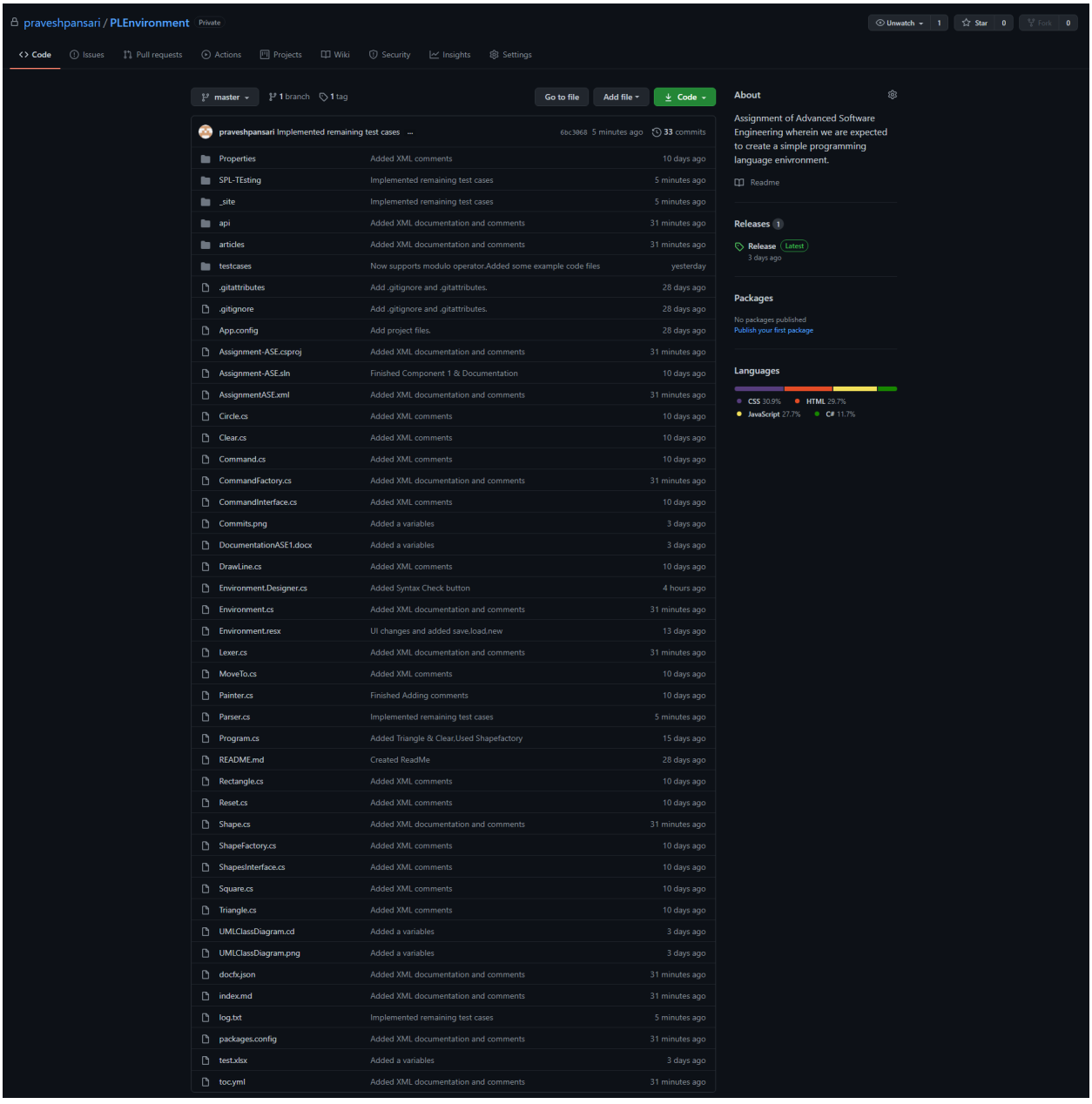
UML Class Diagram



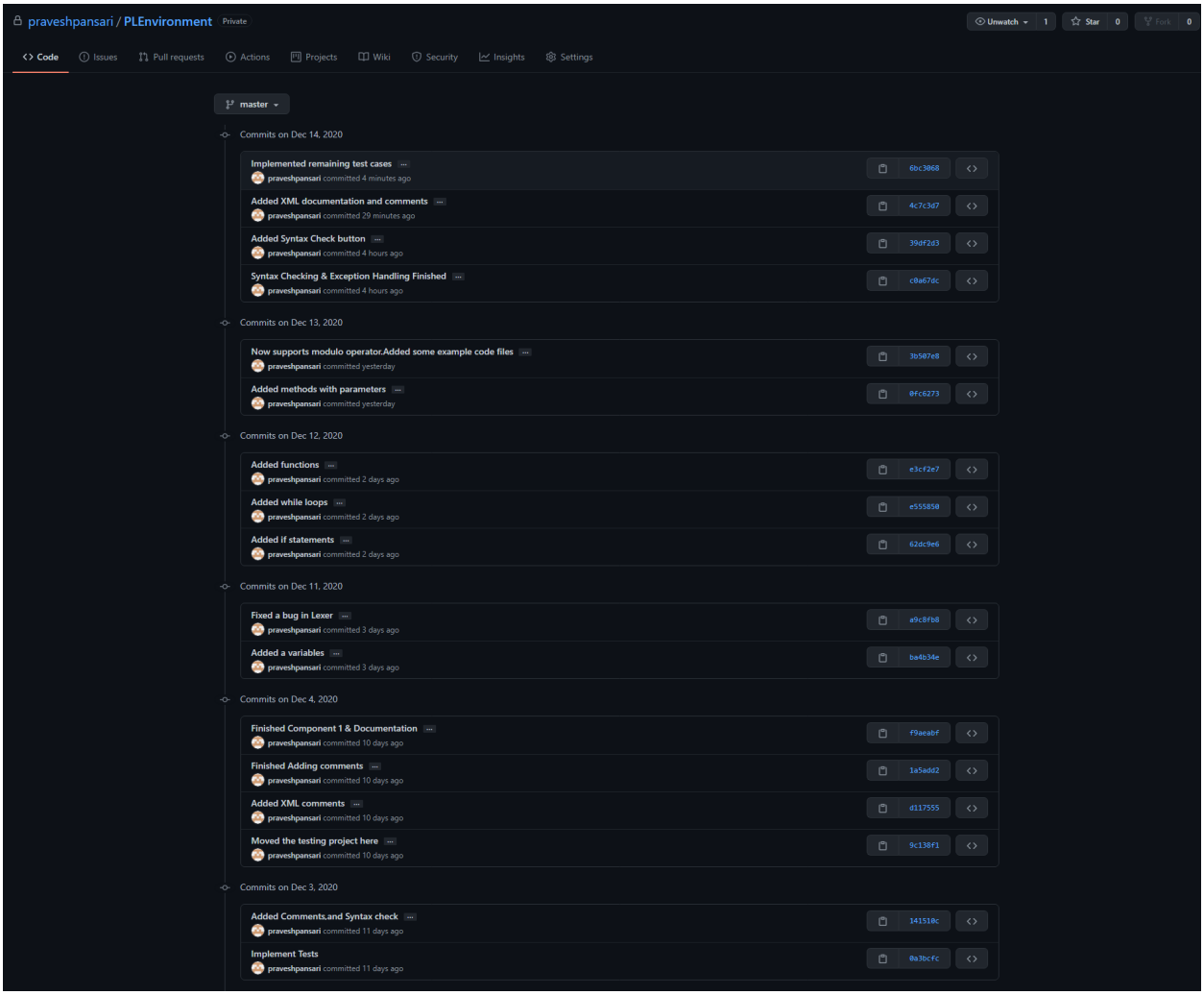
Version Control – GitHub

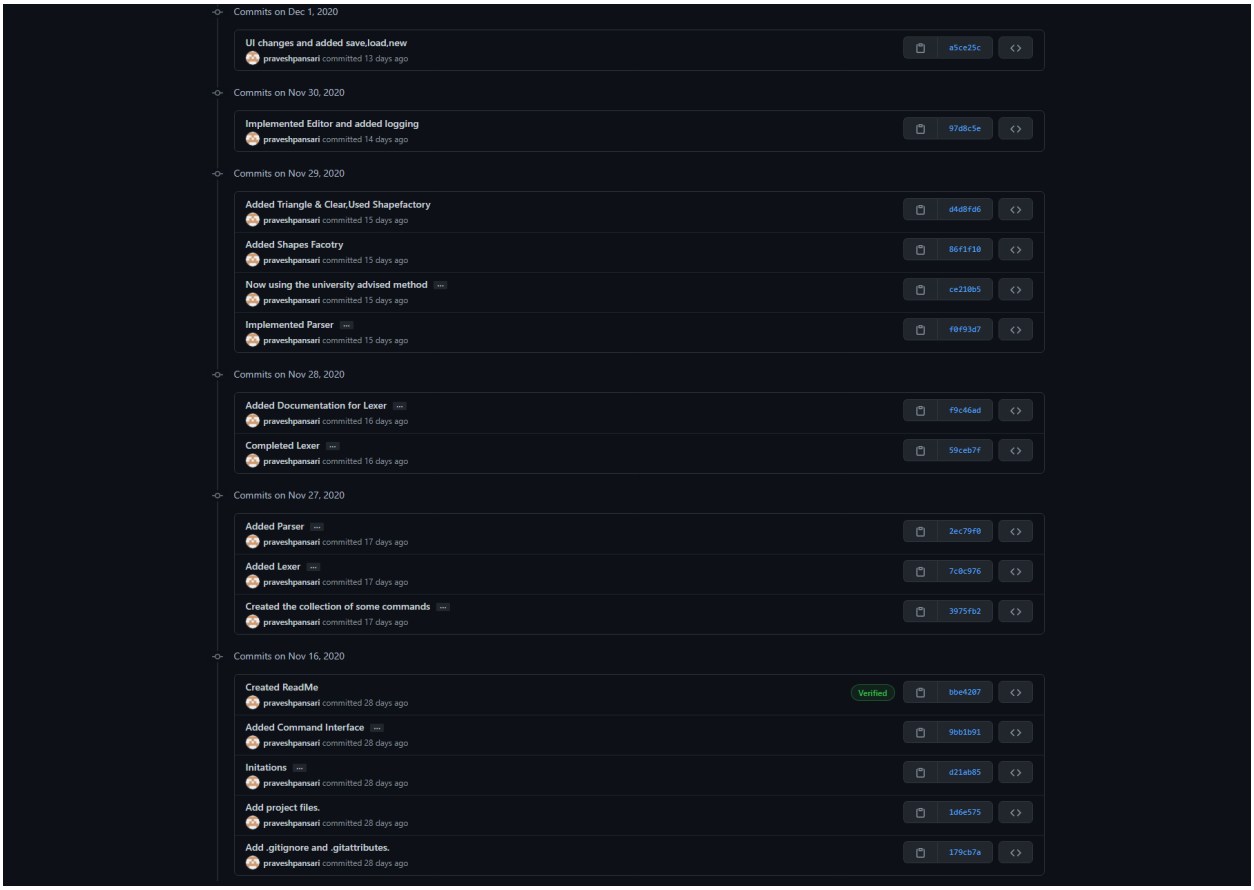
Repository:

Link: https://github.com/praveshpansari/PLEnvironment



Commits:

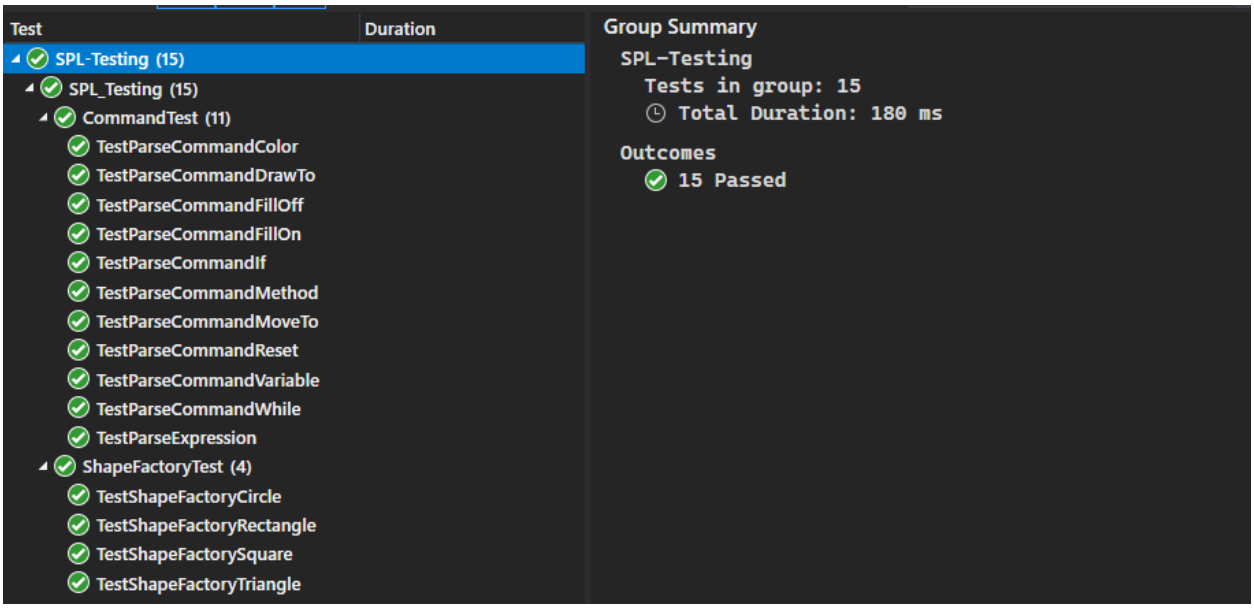

praveshpansari / PLEnvironment
Private
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
master
Commits on Dec 14, 2020
Implemented remaining test cases
praveshpansari committed 4 minutes ago
6bc3068
Added XML documentation and comments
praveshpansari committed 29 minutes ago
4c7c3d7
Added Syntax Check button
praveshpansari committed 4 hours ago
39df2d3
Syntax Checking & Exception Handling Finished
praveshpansari committed 4 hours ago
c0a67dc
Commits on Dec 13, 2020
Now supports modulo operator.Added some example code files
praveshpansari committed yesterday
3b507e8
Added methods with parameters
praveshpansari committed yesterday
0fc6273
Commits on Dec 12, 2020
Added functions
praveshpansari committed 2 days ago
e3cf2e7
Added while loops
praveshpansari committed 2 days ago
e555850
Added if statements
praveshpansari committed 2 days ago
62dc9e6
Commits on Dec 11, 2020
Fixed a bug in Lexer
praveshpansari committed 3 days ago
a9c8fb8
Added a variables
praveshpansari committed 3 days ago
ba4b34e
Commits on Dec 4, 2020
Finished Component 1 & Documentation
praveshpansari committed 10 days ago
f9aeabf
Finished Adding comments
praveshpansari committed 10 days ago
1a5add2
Added XML comments
praveshpansari committed 10 days ago
d117555
Moved the testing project here
praveshpansari committed 10 days ago
9c138f1
Commits on Dec 3, 2020
Added Comments,and Syntax check
praveshpansari committed 11 days ago
141510c
Implement Tests
praveshpansari committed 11 days ago
0a3bcfc


Commits on Dec 1, 2020
UI changes and added save,load,new
praveshpansari committed 13 days ago
a5ce25c
Commits on Nov 30, 2020
Implemented Editor and added logging
praveshpansari committed 14 days ago
97d8c5e
Commits on Nov 29, 2020
Added Triangle & Clear,Used Shapefactory
praveshpansari committed 15 days ago
d4d8fd6
Added Shapes Facotry
praveshpansari committed 15 days ago
86f1f10
Now using the university advised method
praveshpansari committed 15 days ago
ce210b5
Implemented Parser
praveshpansari committed 15 days ago
f0f93d7
Commits on Nov 28, 2020
Added Documentation for Lexer
praveshpansari committed 16 days ago
f9c46ad
Completed Lexer
praveshpansari committed 16 days ago
59ceb7f
Commits on Nov 27, 2020
Added Parser
praveshpansari committed 17 days ago
2ec79f0
Added Lexer
praveshpansari committed 17 days ago
7c0c976
Created the collection of some commands
praveshpansari committed 17 days ago
3975fb2
Commits on Nov 16, 2020
Created ReadMe
praveshpansari committed 28 days ago
Verified
bbe4207
Added Command Interface
praveshpansari committed 28 days ago
9bb1b91
Initations
praveshpansari committed 28 days ago
d21ab85
Add project files.
praveshpansari committed 28 days ago
1d6e575
Add .gitignore and .gitattributes.
praveshpansari committed 28 days ago
179cb7a

Test Cases

Summary:

All tests pass:



All tests fail:

Results:

Shape Factory Test – Rectangle

Test Case ID	SF_01	Test Case Description	Test for the shape factory		
Created By	Pravesh Pansari			Version	1.0
Tester's Name	Pravesh Pansari	Date Tested	4th December,	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Shape Interface and Shape classes
3	Access to Shape Factory
4	

S #	Test Data
1	GetShape parameter = "rect"
2	
3	
4	

Test Scena: Verify whether the required shape can be generated from the factory

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a ShapeFactory object	ShapFactory Object	As Expected	Pass
2	Create a Shape object	Shape object created	As Expected	Pass
3	Use the getShape() Method	Rectangle Shape	As Expected	Pass

Shape Factory Test – Triangle

Test Case ID	SF_02	Test Case Description	Test for the shape factory		
Created By	Pravesh Pansari			Version	1.0
Tester's Name	Pravesh Pansari	Date Tested	4th December,	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Shape Interface and Shape classes
3	Access to Shape Factory
4	

S #	Test Data
1	GetShape parameter = "triangle"
2	
3	
4	

Test Scena: Verify whether the required shape can be generated from the factory

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a ShapeFactory object	ShapFactory Object	As Expected	Pass
2	Create a Shape object	Shape object created	As Expected	Pass
3	Use the getShape() Method	Triangle Shape Created	As Expected	Pass

Command Test – Move to

Test Case ID	C_01	Test Case Description	Test for Move to command		
Created By	Pravesh Pansari			Version	1.0
Tester's Name	Pravesh Pansari	Date Tested	4th December,	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Command Interface, Command
3	Access to Command Factory
4	Access to Parser,Painter classes

S #	Test Data
1	parseCommand(input,0), where input = "moveto
2	
3	
4	

Test Scena: Verify whether the x position and the y position of the cursor have been moved

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use the parseCommand() method	Cursor moved to	As Expected	Pass

Command Test – Draw To

Test Case ID	C_02	Test Case Description	Test for Draw to command		
Created By	Pravesh Pansari			Version	1.0
Tester's Name	Pravesh Pansari	Date Tested	4th December,	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Command Interface, Command
3	Access to Command Factory
4	Access to Parser,Painter classes

S #	Test Data
1	parseCommand(input,0), where input = "drawto
2	
3	
4	

Test Scena: Verify whether the x position and the y position of the cursor have been moved

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use the parseCommand() method	Cursor moved to	As Expected	Pass

Command Test – Reset Command

Test Case ID	C_03	Test Case Description	Test for reset command		
Created By	Pravesh Pansari			Version	1.0
Tester's Name	Pravesh Pansari	Date Tested	4th December,	Test Case (Pass/Fail)	Pass

S #	Prerequisites:	S #	Test Data
1	Access to Visual Studio 2019	1	parseCommand(input,0), where input = "reset"
2	Access to Command Interface, Command	2	
3	Access to Command Factory	3	
4	Access to Parser,Painter classes	4	

Test Scena: Verify whether the x position and the y position of the cursor have been set to top left

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use the parseCommand() method	Cursor reset to top left	As Expected	Pass

Command Test – Pen Command

Test Case ID	C_04	Test Case Description	Test for pen command		
Created By	Pravesh Pansari			Version	1.0
Tester's Name	Pravesh Pansari	Date Tested	4th December,	Test Case (Pass/Fail)	Pass

S #	Prerequisites:	S #	Test Data
1	Access to Visual Studio 2019	1	parseCommand(input,0), where input = "pen red"
2	Access to Command Interface, Command	2	
3	Access to Command Factory	3	
4	Access to Parser,Painter classes	4	

Test Scena: Verify whether the pen color has been set to red

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use the parseCommand() method	Pen color set to red	As Expected	Pass

Command Test – Create Variable

Test Case ID	C_05	Test Case Description	Test for var command		
Created By	Pravesh Pansari			Version	1.1
Tester's Name	Pravesh Pansari	Date Tested	14th December, 2020	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Command Interface, Command
3	Access to Command Factory
4	Access to Parser,Painter,Lexer classes

S #	Test Data
1	parseUsingLexer("num = 5",0)
2	
3	
4	

Test Scena: Verify whether the pen color has been set to red

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use parseUsingLexer() method	Create variable "num" with value 5	As Expected	Pass

Command Test – While

Test Case ID	C_06	Test Case Description	Test for while command		
Created By	Pravesh Pansari			Version	1.1
Tester's Name	Pravesh Pansari	Date Tested	14th December, 2020	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Command Interface, Command
3	Access to Command Factory
4	Access to Parser,Painter, Lexer classes

S #	Test Data
1	parseEditor(input) where input is "num = 5\r\nwhile num > 3\r\nnum = num - 1\r\nendwhile"

Test Scena: Verify whethernum has been decreased to 3 from 5

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use parseEditor() method	Num decrease to 3	As Expected	Pass

Command Test – If statement

Test Case ID	C_07	Test Case Description	Test for if command		
Created By	Pravesh Pansari			Version	1.1
Tester's Name	Pravesh Pansari	Date Tested	14th December, 2020	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Command Interface, Command
3	Access to Command Factory
4	Access to Parser,Painter, Lexer classes

S #	Test Data
1	parseUsingIf(input,0) where input is "if 5 > 20"

Test Scena: Verify whether 5 is greater than 20 in if statement returns false

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use parseUsingIf() method	Returns false	As Expected	Pass

Command Test – Create Method

Test Case ID	C_08	Test Case Description	Test for method command		
Created By	Pravesh Pansari			Version	1.1
Tester's Name	Pravesh Pansari	Date Tested	14th December, 2020	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Command Interface, Command
3	Access to Command Factory
4	Access to Parser,Painter, Lexer classes

S #	Test Data
1	parseEditor(input) where input is "method example()\r\ncircle 20\r\nendmethod"

Test Scena: Verify whether method example has been created

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use parseEditor() method	Method created example	As Expected	Pass

Command Test – Complex Expression

Test Case ID	C_09	Test Case Description	Test for expressions		
Created By	Pravesh Pansari			Version	1.1
Tester's Name	Pravesh Pansari	Date Tested	14th December, 2020	Test Case (Pass/Fail)	Pass

S #	Prerequisites:
1	Access to Visual Studio 2019
2	Access to Command Interface, Command
3	Access to Command Factory
4	Access to Parser,Painter, Lexer classes

S #	Test Data
1	parseUsingLexer(input,0) where input is "x = 15"
2	parseUsingLexer(input,0) where input is "num = 5 + x *2"

Test Scena Verify whether num has been set to 40

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Create a Painter Object	Painter object created	As Expected	Pass
2	Create a Parser Object	Parser object created	As Expected	Pass
3	Use parseUsingLexer() method	Variable x created	As Expected	Pass
4	Use parseUsingLexer() method	Variable num created with value 40	As Expected	Pass