

COMS W4115

# Programming Languages & Translators

## NewLogo Team

Shulei Zhao(Project Manager)

Pengfei Weng(Language and Tools Guru)

Yuandi Jin(System Architect)

Adarsh Banad(System Integrator)

Yang Pan(Tester and Validator)



# Topic & Motivation

- We may need illustration and graphics everyday.  
Office Studio Convenient but not accurate  
Java or Matlab Accurate but not convenient
- NewLogo focuses on the Drawing process rather than the mathematical expression.
- Making it more convenient as well as accurate



# Features

- **Easy to learn**
- **More Freedom**
  - **Hand-painting experiences**
- **More Accuracy**
  - **Control with parameters**



# Language Tutorial

- o **Basic Data Types**
  - o int, double, boolean, String
- o **Loops**
  - o For-loop and While-loop
- o **Conditions**
  - o If and If-Else



# Language Tutorial

- **Comments**

- **# this is a comment**

- **Function**

- **Exactly one void main function**

- **Zero or more user-defined functions**

# Language Tutorial

## ○ Drawing Functionality

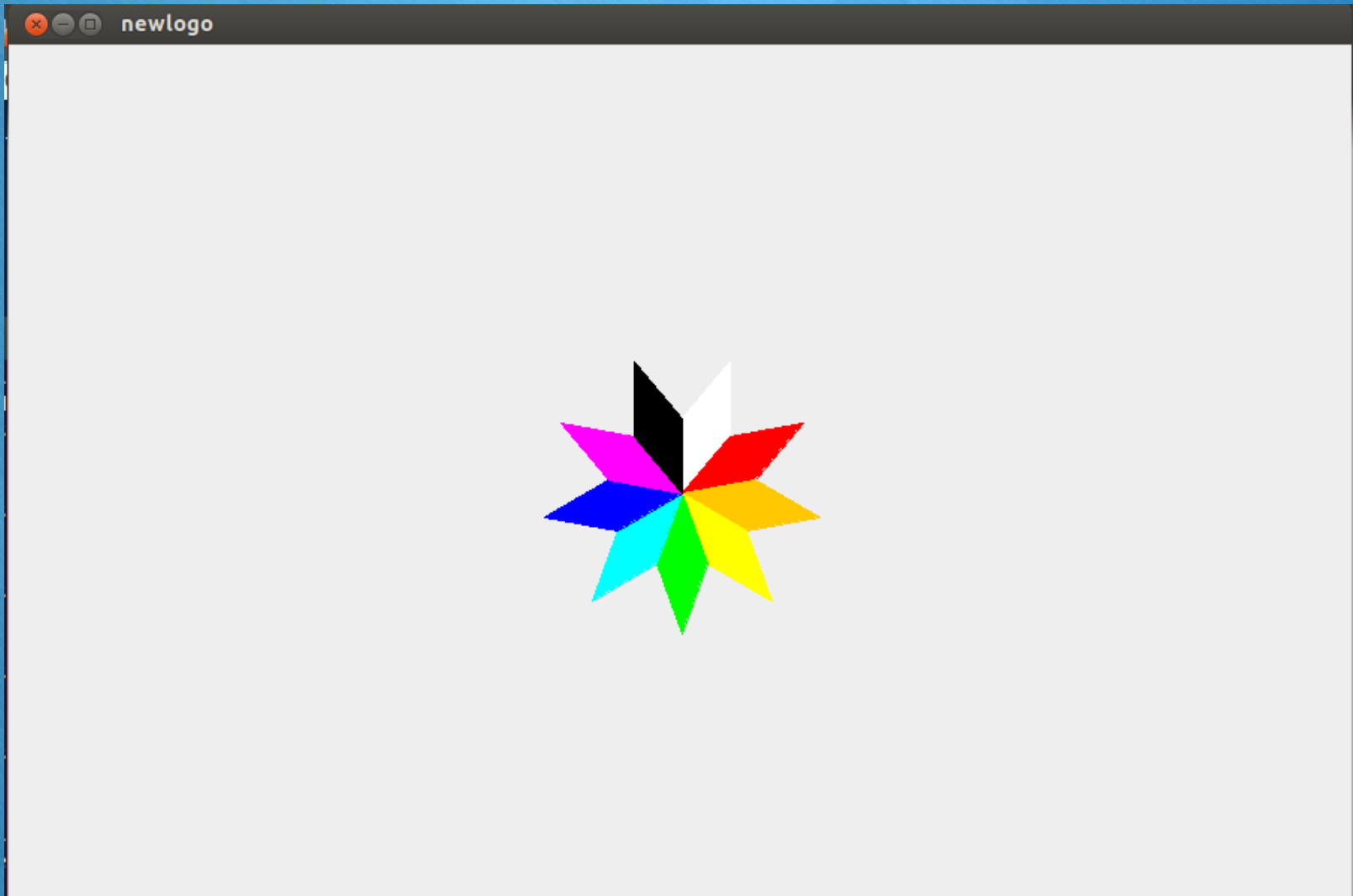
<code>forward int</code>	move forward int steps
<code>back int</code>	move back int steps
<code>right int/double</code>	turn right int degrees
<code>left int/double</code>	turn left int degrees
<code>pup</code>	raise pen up, moving without drawing
<code>pdown</code>	put pen down, ready for drawing
<code>pcolor int</code>	change the color of the pen, we provide 9 different colors
<code>pwidth int</code>	change the width of the lines drawn by the pen
<code>fillstart/fillend</code>	marks the starting/ending point to fill some geometry
<code>print String</code>	print a string to the screen in the current pen position

# Sample Program

```
void drawFlower(int n){
    for(int i = 0; i < n; i++){
        pcolor i;# draw different petals with different colors
        fillstart;# prepare to fill the petal
        forward 50;
        right 40;
        forward 50;
        right 140;
        forward 50;
        right 40;
        forward 50;
        right 140;# draw a diamond to represent the petal
        fillend;# fill the petal
        right 40;
    }
}

void main ( ) {
    drawFlower(9);# call the function to draw the flower
}
```

# Sample Program







# Translator Architecture

- In NewLogo, we have essentially 5 modules: scanner, parser/ast, semantic, generator and Java code frame.
- Our compiler takes as input a source program in NewLogo language (\*.nlg), and outputs a Java program (\*.java).

source program

Scanner

tokens

Parser/ast

AST

Semantic Analyzer

correct AST

Generator

code fragment

Java Code Frame

NewLogo Compiler

Java program

Java Translator

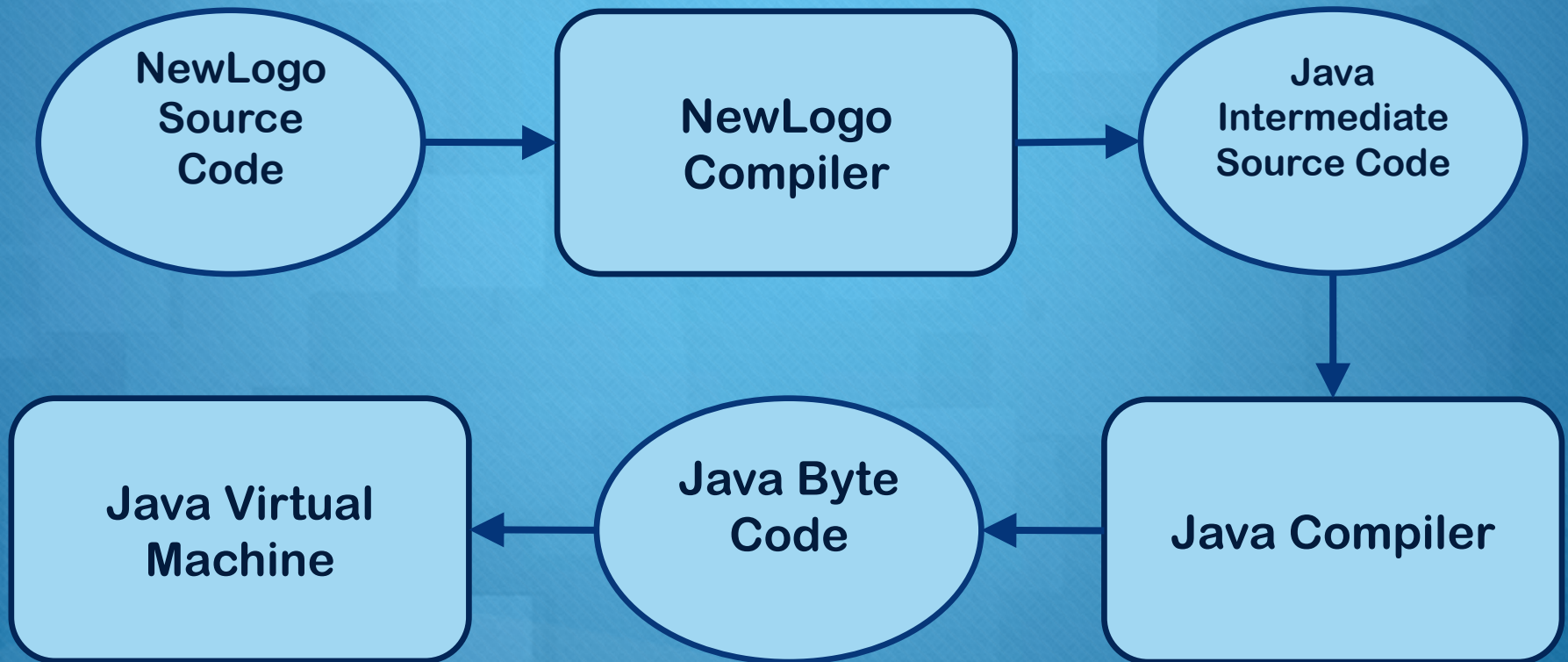
Java bytecode

JVM

result

<b>Module</b>	<b>Files</b>	<b>Written by</b>	<b>Modified by</b>
<b>Scanner</b>	header.ml	Pengfei Weng	
	scanner.mli	Pengfei Weng	
<b>Parser/ast</b>	parser.mly	Pengfei Weng	Yuandi Jin, Adarsh Banad
	ast.ml	Pengfei Weng	Yuandi Jin, Adarsh Banad
	parser.mli	Pengfei Weng	
<b>Semantic Analyzer</b>	semantic.ml	Yang Pan, Adarsh Banad	All other members
	semantic.mli	Yang Pan, Adarsh Banad	
<b>Generator</b>	generator.ml	Yuandi Jin, Adarsh Banad	
	generator.mli	Yuandi Jin, Adarsh Banad	
<b>Java code frame</b>	NewLogoPanel (copy).java	Shulei Zhao	
	NewLogoFrame .java	Shulei Zhao	

# Run-time Environment



# Development Environment

## ◦ Eclipse

- Development and testing



## ◦ Dropbox

- Code repository

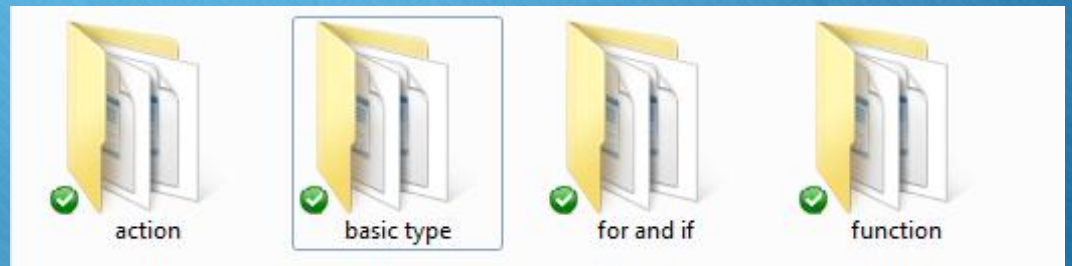
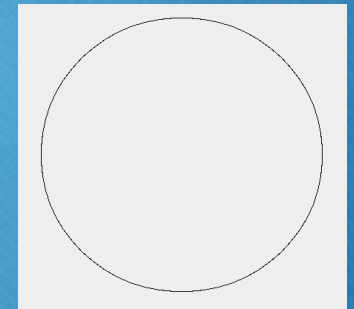
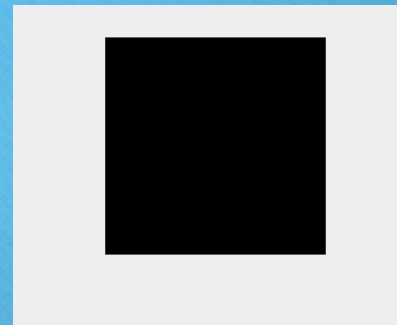
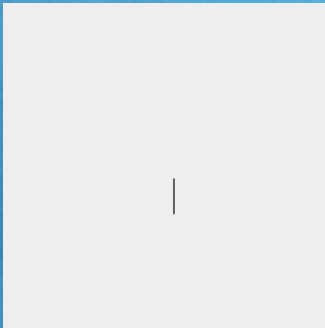


**Dropbox**

# Test Cases

## ○ Simple Function Tests

- `void main () {`
- `int a = 20;`
- `forward a;`
- `}`



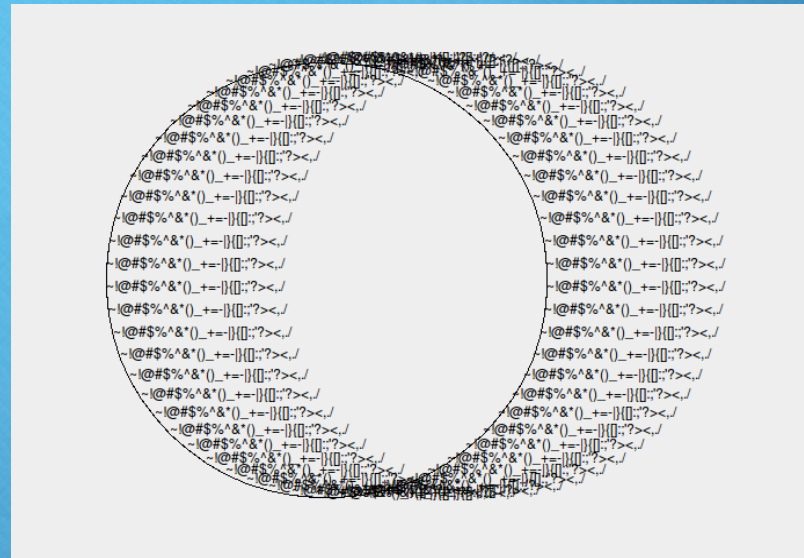
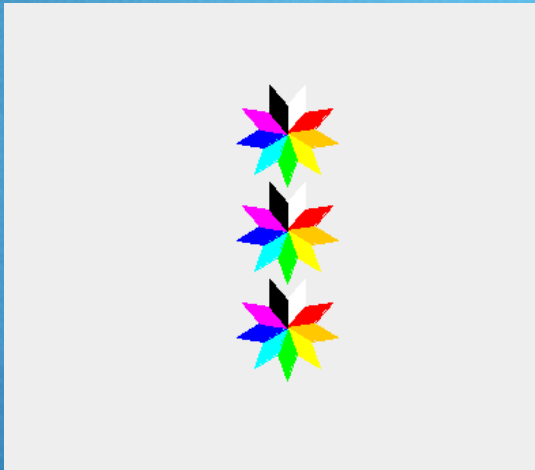
# Test Cases

- The result of semantic checking:

```
./compiler -c ./error001.nlg ./NewLogoPanel.java
Semantic Error: Undeclared variable 'i' at line 3
Semantic Error: Action 'fillstart' must has parameter
Semantic Error: Action 'fillend' must has parameter
Semantic Error: no operation defined for type 'int' at line 2
Semantic Error: no operation defined for type 'int' at line 20
javac NewLogoFrame.java
java NewLogoFrame
```

# Test Cases

## Complex Test Case







# Lessons Learned

- **What we learned**
  - **Work as a team but not five individuals**
  - **Remember the roles but do not be limited by the roles**
- **What worked well**
  - **Clear Destination**
  - **Team Programming**



# Advice for Future Teams

- **Start Early – The scope and complexity of this project is likely greater than any you have done before.**
- **Keep the Team Agile – Don't feel limited by your given responsibilities; take on additional work and help out your team members as needed.**
- **Try New Tools – Don't limit yourself to the tools learned in class.**

A decorative header featuring a bright yellow sun with a blue circle in the center, partially obscured by blue and white stylized clouds. The background is a solid blue color with a subtle grid pattern.

# Q&A

A stylized illustration of a bright yellow sun with a blue circle in the center, partially obscured by blue and white clouds. The background is a solid blue color with a subtle pattern of lighter blue squares.

**Thank YOU !**