

## Objectives

- SLogo Design

Nov 19, 2008

Sprenkle - CS209

1

## SLogo Sketches

- Draw your SLogo Interfaces
- Map to your classes, as appropriate
- List other associated non-GUI classes

Nov 19, 2008

Sprenkle - CS209

2

## Analyzing SLogo Sketches

- Looking for common classes/sketches/design patterns

Nov 19, 2008

Sprenkle - CS209

3

## Analysis

- Use Case
  - The user starts the program, types 'fd 50' in the command window, and sees the turtle move in the display window leaving a trail.

Nov 19, 2008

Sprenkle - CS209

4

## SLogo Language

- SLogo has a lot of commands
- When a user enters a command, the application needs to execute the command
- Examples
  - FORWARD 50
  - RIGHT 90
  - REPEAT numOrVar [ instructionList ]

Nov 19, 2008

Sprenkle - CS209

5

## Programming Language Syntax

- What does an identifier look like in Java?
- What does an assignment statement look like in Java?
- What can be on the left hand side?
- What can be on the right hand side?
- What does a multiplication look like?
- How do we evaluate arithmetic expressions?

Nov 19, 2008

Sprenkle - CS209

6

## Programming Language Design

- Must be unambiguous
  - Programming Language defines a syntax and semantics
- Interpreting programming languages
  - Parse program into tokens
    - Example: `x = 4*3;` →  
`<id> <assignment> <num> <mult> <num> <endofstmt>`
  - Validate that tokens are in a valid form
  - Generate executable code

Nov 19, 2008

Sprenkle - CS209

7

## A Grammar To Describe Expressions

- Left-recursive grammar
 

```
<Exp> ::= <Exp> + <Term> |
        <Exp> - <Term> |
        <Term>

<Term> ::= <Term> * <Factor> |
          <Term> / <Factor> |
          <Factor>

<Factor> ::= number | <id> |
           ( <Exp> ) |
           - <Factor>
```

Nov 19, 2008

Sprenkle - CS209

8

## SLogo Starting Code

- Import an existing project from
  - `/home/courses/cs209/handouts/slogo.tar`

Nov 19, 2008

Sprenkle - CS209

9

## Understanding Given Code

- What code is a very basic GUI for SLogo?
  - How do you execute the code?
  - How is the code organized?
  - What design pattern does it match?

Nov 19, 2008

Sprenkle - CS209

10

## Understanding the Given Code

- What packages contain a basic LL(1) language parser?

Nov 19, 2008

Sprenkle - CS209

11

## Packages/Classes to Know

- `jeJan.simple`
  - Simple parser of expressions
  - Mixes together tokens, expressions, parsers
- Tokens
  - `CharTokens` – single-character tokens
  - `IdentifierToken` – represents an identifier
  - `NumberToken` – represents a number
- Factory: `CharTokensFactory`
- Expressions
  - Binary, Unary
  - Example: Addition
- Parser: `ELanParser`
  - Run on `tests/basic_expressions2`

Nov 19, 2008

Sprenkle - CS209

12

## ElanParser Close Up

- StreamTokenizer customization
- Parser object
- Parsing the stream

Nov 19, 2008

Sprenkle - CS209

13

## For Friday: Project Preparation

- Read over the SLogo (Final Project) specifications again
  - More than on the take-home exam question
- First deliverable is a text document that answers
  - What needs to be completed?
  - What is your plan for completing those tasks?
  - What tasks are you most interested in working on?
  - ....
- Friday
  - Discuss your plans, questions
  - Discuss tools to help collaboration

Nov 19, 2008

Sprenkle - CS209

14