Cardigan
Card Game Development

Joshua Lopez (Team Leader)
Muzi Gao  *  Miriam Melnick
Introduction

What is Cardigan?
Here's a hint, it's not a sweater.

- Card game development & implementation language
- Built in data types which support card game elements (cards, players, etc.)
- Control structures for game play (rules, turns, winning conditions, etc.)
Motivation

- Developing card games is tedious
- Materials are expensive
- Physical iteration takes time
Motivation

Developing in code allows

- no cost for materials
- easy modification
- fast iteration
- better development
Lexical Analysis converts source file to tokens

suits = ['hearts', 'clubs', 'diamonds', 'spades']
ranks = ['a', 'k', 'q', 'j', '10', '9', '8', '7', '6', '5', '4', '3', '2']

PLAY()=
    deck = cartesian(suits, ranks)
    player = {'name': '', 'score': 0, 'hand': []}

ID ASSIGN LBRAC ID COMMA ID COMMA ID COMMA ID RBRAC EOL ID
ASSIGN LBRAC STRING COMMA STRING COMMA STRING COMMA STRING COMMA STRING COMMA STRING COMMA STRING COMMA STRING COMMA STRING COMMA STRING RBRAC EOL ID
LPAREN RPAREN ASSIGN LCURL ID ASSGN ID LPAREN ID COMMA ID RPAREN EOL ID ASSIGN LCURL ID COLON STRING COMMA ID COLON INT COMMA ID COLON LBRAC RBRAC RCUMLRCURL EOL ...

scanner.mll
Syntactic analysis creates an abstract syntax tree
Compiling an AST in 3 stages

- **Separator**
  - Breaks off subtrees

- **Semantic analysis**
  - Keeps track of types in a symbol table
  - Checks each subtree to make sure types are valid

- **Code generation**
  - Creates Java code from templates
Implementation

- AST Definitions
- Cardigan Source
- Scanner
  - Tokens
  - Lexical Analysis
- Parser
  - Syntactic Analysis
  - A. S. T.
- Translator
  - Semantic Analysis
  - Code Generation
- Javac
  - Java Source
  - Java Compilation
- JVM
  - Java Bytecode
  - Execution
- hello cardigan

Standard Library
Lessons Learned

- Newline characters are not good line delimiters
- Type inference is hard
- Ocaml is hard to debug
- It's possible to start too early
- Work with "real" code as well as tests the whole time