maze

Alexander Brown (aab2212)
Alexander Freemantle (asf2161)
Michelle Navarro (mn2614)
Lindsay Schiminske (ls3245)

Introduction

What is maze?

Language Evolution

Proposal:

Game design language

Plan:

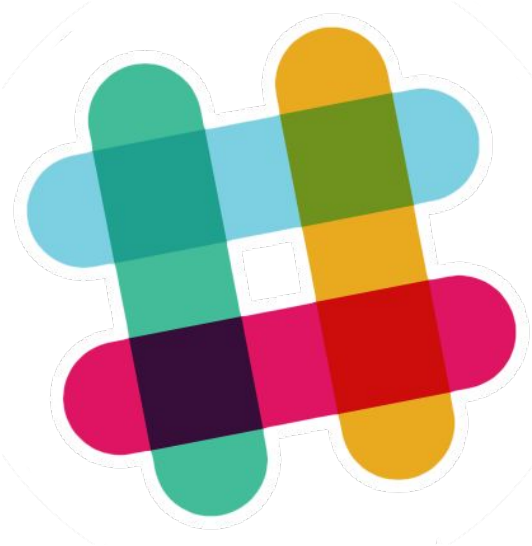
Object-Oriented Java-like language

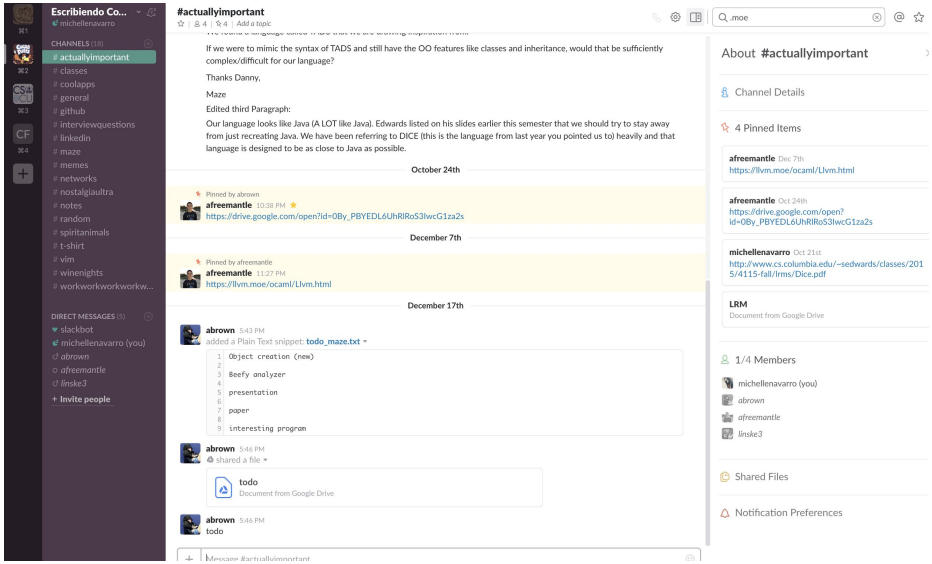
Actual:

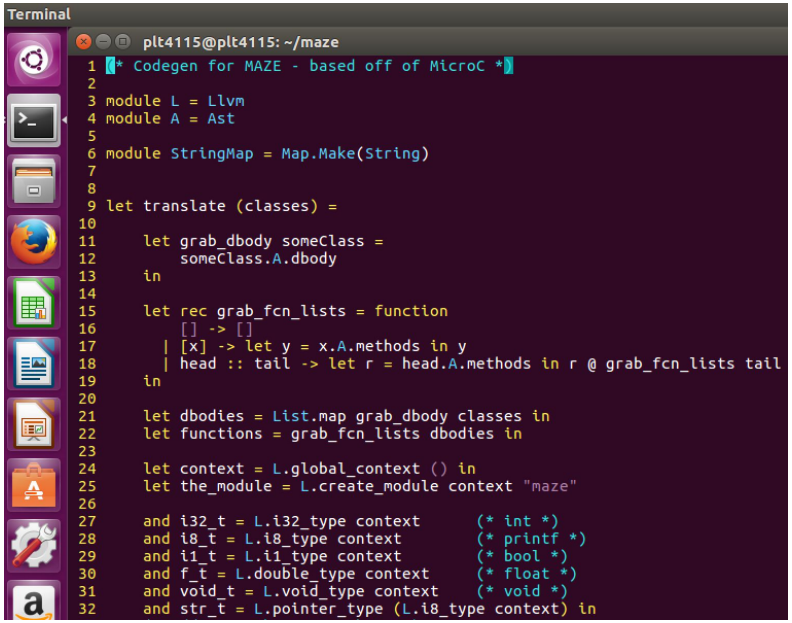
Java-like imperative language

An Imperative Language with Java-like syntax

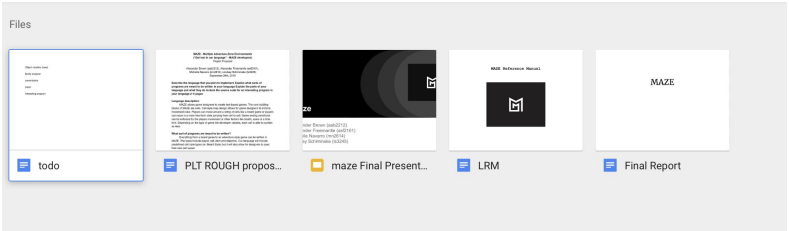
Tools



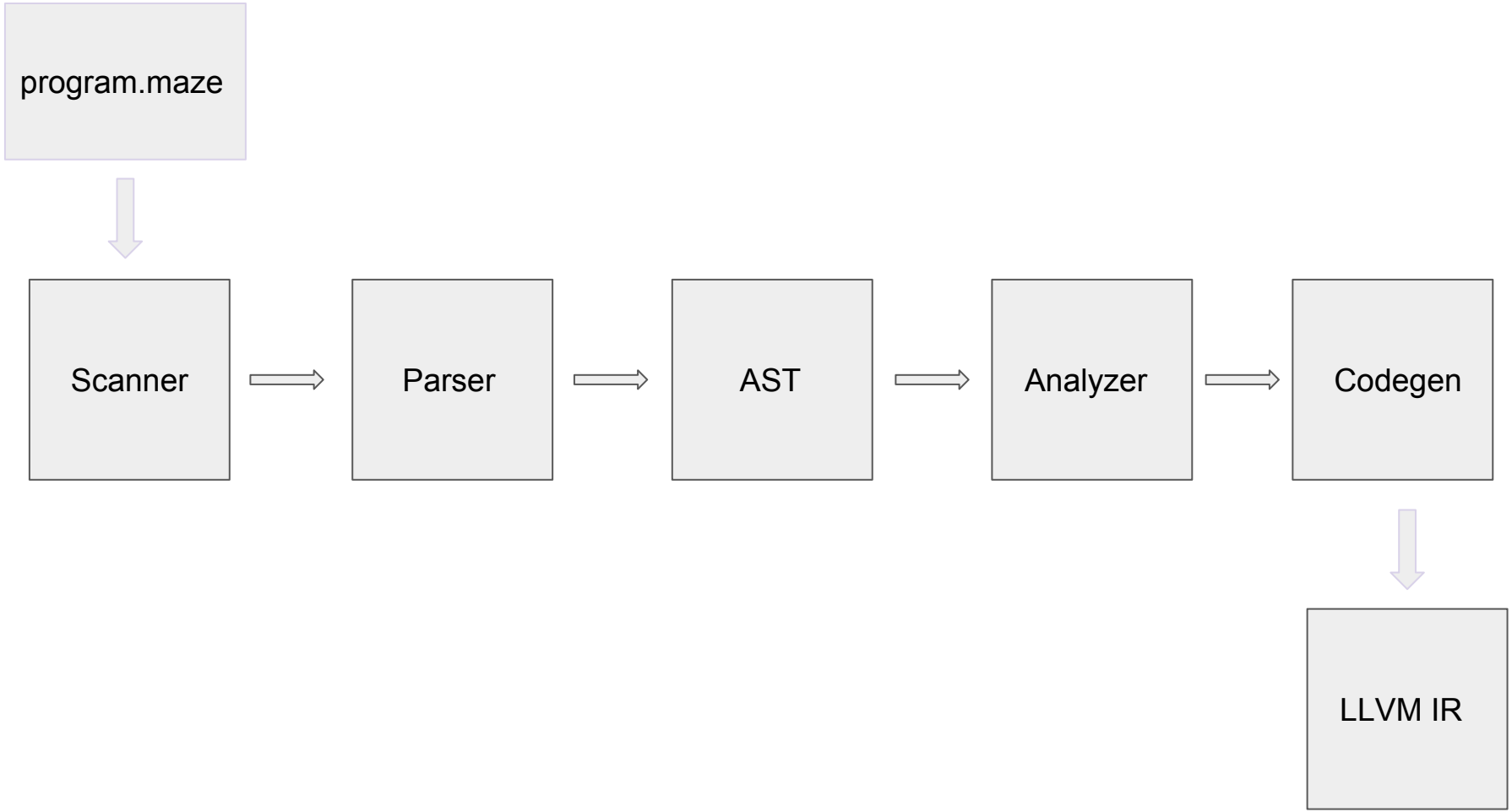

Escribiendo Co...
michellenavarro
CHANNELS (18)
actuallyimportant
classes
coolapps
general
github
interviewquestions
linkedin
maze
memes
networks
nostalgiaultra
notes
random
spiritanimals
t-shirt
vim
winenights
workworkworkworkw...
DIRECT MESSAGES (5)
slackbot
michellenavarro (you)
abrown
afreemantle
linske3
+ Invite people
#actuallyimportant
If we were to mimic the syntax of TADS and still have the OO features like classes and inheritance, would that be sufficiently complex/difficult for our language?
Thanks Danny,
Maze
Edited third Paragraph:
Our language looks like Java (A LOT like Java). Edwards listed on his slides earlier this semester that we should try to stay away from just recreating Java. We have been referring to DICE (this is the language from last year you pointed us to) heavily and that language is designed to be as close to Java as possible.
October 24th
Pinned by abrown
afreemantle 10:38 PM
https://drive.google.com/open?id=0By_PBYEDL6UhRlRoS3lwcG1za2s
December 7th
Pinned by afreemantle
afreemantle 11:27 PM
https://llvm.moe/ocaml/Llvm.html
December 17th
abrown 5:43 PM
added a Plain Text snippet: todo_maze.txt
Object creation (new)
Beefy analyzer
presentation
paper
interesting program
abrown 5:46 PM
shared a file
todo
Document from Google Drive
abrown 5:46 PM
todo
Message #actuallyimportant
About #actuallyimportant
Channel Details
4 Pinned Items
afreemantle Dec 7th
https://llvm.moe/ocaml/Llvm.html
afreemantle Oct 24th
https://drive.google.com/open?id=0By_PBYEDL6UhRlRoS3lwcG1za2s
michellenavarro Oct 21st
http://www.cs.columbia.edu/~sedwards/classes/2015/4115-fall/lrms/Dice.pdf
LRM
Document from Google Drive
1/4 Members
michellenavarro (you)
abrown
afreemantle
linske3
Shared Files
Notification Preferences


Terminal
plt4115@plt4115: ~/maze
(* Codegen for MAZE - based off of MicroC *)
module L = Llvm
module A = Ast
module StringMap = Map.Make(String)
let translate (classes) =
let grab_dbody someClass =
someClass.A.dbody
in
let rec grab_fcn_lists = function
[] -> []
| [x] -> let y = x.A.methods in y
| head :: tail -> let r = head.A.methods in r @ grab_fcn_lists tail
in
let dbodies = List.map grab_dbody classes in
let functions = grab_fcn_lists dbodies in
let context = L.global_context () in
let the_module = L.create_module context "maze"
and i32_t = L.i32_type context (* int *)
and i8_t = L.i8_type context (* printf *)
and i1_t = L.i1_type context (* bool *)
and f_t = L.double_type context (* float *)
and void_t = L.void_type context (* void *)
and str_t = L.pointer_type (L.i8_type context) in


Oct 2, 2016 – Dec 19, 2016
Contributions to master, excluding merge commits
Contributions: Commits
60
40
20
0
Oct 02
Oct 09
Oct 16
Oct 23
Oct 30
Nov 06
Nov 13
Nov 20
Nov 27
Dec 04
Dec 11
Dec 18


Files
todo
PLT ROUGH propos...
maze Final Present...
LRM
MAZE
Final Report

MAZE Architecture


program.maze
Scanner
Parser
AST
Analyzer
Codegen
LLVM IR

Syntax

Basics

Primitives

```
int
float
bool
char
```

```
string
```

Binary Ops

```
+ - * /
< <= > >=
== != && ||
```

Unary Ops

```
! -
```

Comments

```
(* comment *)
```

Methods

```
class test {

    void dummy(){
        int x;
        x = 1;
        print(x);
    }

    void main(){
        dummy();
    }
}
```

Statements

If / else

```
class test{
    void main(){
        int x;
        x = 0;

        if(x>1){
            x = x - 1;
        }
        else if(x == 1){
            x = x * 3;
        }
        else{
           x = x + 5;
           print(x);
        }
    }
}
```

Return

```
class test {

    int computeValue(int x, int y){
        return x + y;
    }

    void main(){

        int a;
        a = computeValue(10,5);
        print(a);
    }
}
```

While Loop

```
class test {
    void main () {

        int i;
        i = 0;

        while(i < 5){
            print("while loop");
            i = i + 1;
        }
    }
}
```

Testing

Menhir

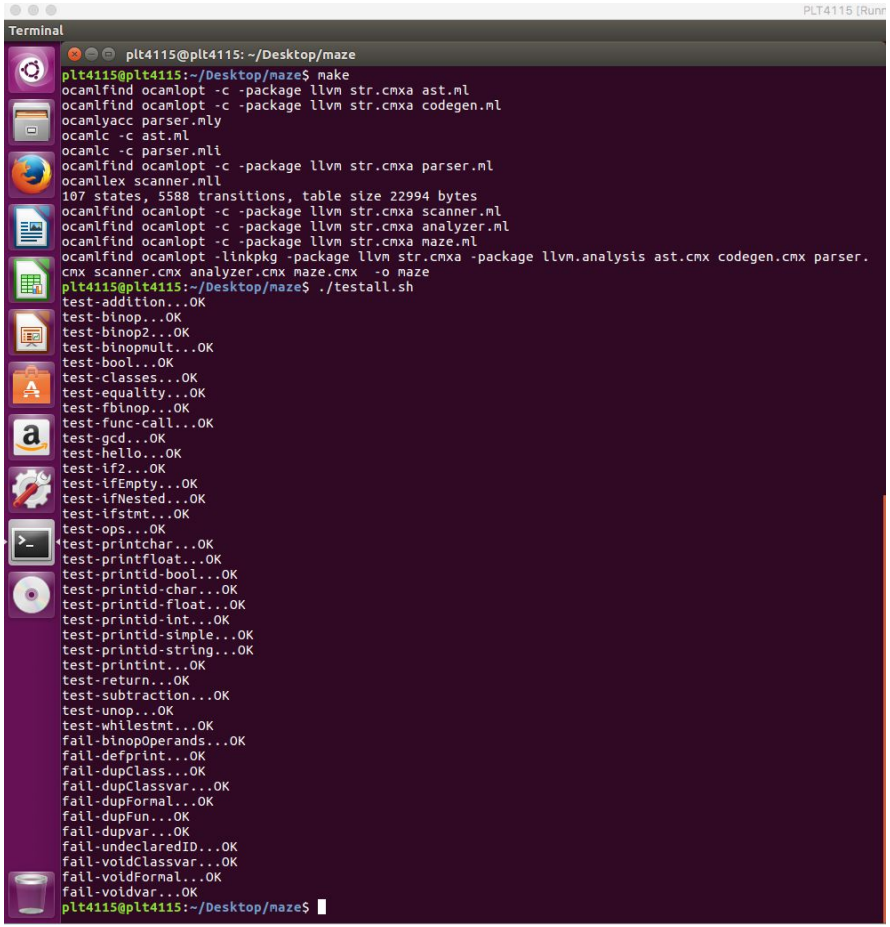
$ menhir --interpret --interpret-show-cst parser.mly

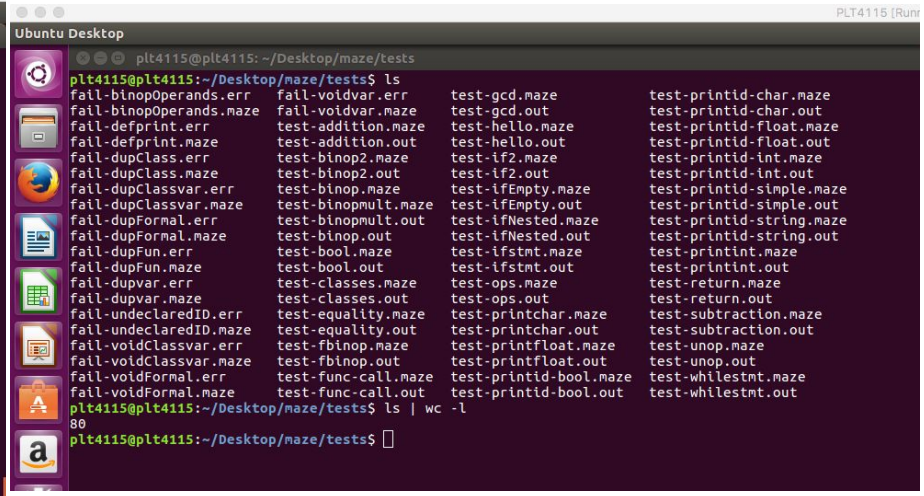
--explain

```
CLASS ID LBRACE INT ID LPAREN RPAREN LBRACE RBRACE RBRACE
ACCEPT
[program:
  [decls:
    [class_decl_list:
      [class_decl:
        CLASS
        ID
        LBRACE
        [dbody:
          [dbody:]
          [fdecl:
            [typ: INT]
            [fname: ID]
            LPAREN
            [formals_opt:]
            RPAREN
            LBRACE
            [vdecl_list:]
            [stmt_list:]
            RBRACE
          ]
        ]
        RBRACE
      ]
    ]
  ]
  EOF
]
```

Pretty-Printer

Made sure input tokens = output tokens





- Add feature → Add test
- Run ./testall.sh
- Ensure all tests pass

Tests that should pass:

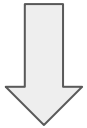
plt4115@plt4115: ~/maze/tests

```
1  class test {
2      void main () {
3
4          int x;
5          x = 3 + 7;
6
7          if(x==10){
8              print("correct");
9          }
10     }
11 }
```

Tests that should fail (with appropriate error message)

plt4115@plt4115: ~/maze/tests

```
1  class test {
2      void main () {
3          int x;
4          int x;
5      }
6  }
```



```
plt4115@plt4115:~/maze$ ./maze -c tests/fail-dupvar.maze
Fatal error: exception Failure("duplicate variable x")
```

GCD

plt4115@plt4115: ~/maze/tests

```
 1 class test {
 2     void main () {
 3         int x;
 4         int y;
 5         x = 15;
 6         y = 20;
 7
 8         if (x == 0) {
 9            print(x);
10         }
11
12         while(x != y)
13         {
14             if ( x > y) {
15                 x= x-y;
16             } else {
17                 y= y-x;
18             }
19         }
20         print(x);
21     }
22 }
~
~
~
~
"test-gcd.maze" 22L, 334C                    1,1
```

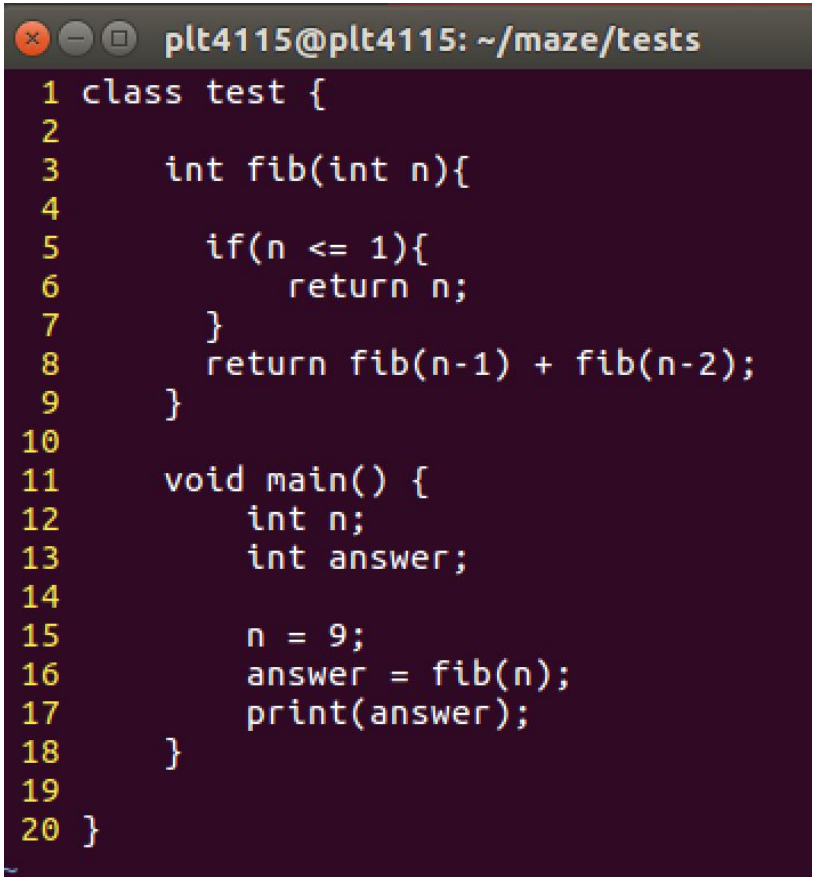
Passing print an identifier:

plt4115@plt4115: ~/maze/tests

```
1 class test {
2
3     void main(){
4         string x;
5         x = "Hello";
6         print(x);
7     }
8
9 }
```

DEMO

Fibonacci is cool


plt4115@plt4115: ~/maze/tests
1 class test {
2
3 int fib(int n){
4
5 if(n <= 1){
6 return n;
7 }
8 return fib(n-1) + fib(n-2);
9 }
10
11 void main() {
12 int n;
13 int answer;
14
15 n = 9;
16 answer = fib(n);
17 print(answer);
18 }
19
20 }