Words

COMS 4115 Final Project Presentation

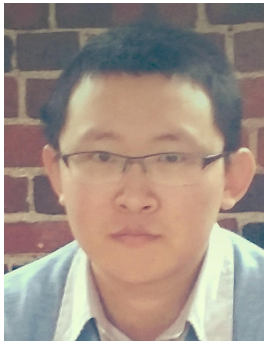
Andrew Kallem
Project Manager

Alex Mark
Language Guru

James Young
System Architect

Michael Ben-Ami
System Integrator

Wangda Zhang
Tester



Overview

Hello World?

```
#include <stdio.h>

int main(int argc, char **argv) {
    printf("Hello World");
    return 0;
}
```

Hello World?

```
#include <st

int main(int
    printf("H
    return 0;
}
```

Hello World???

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Hello World???

```
class HelloW
    public s                    args) {
        Syst                   rld");
    }
}
```

Hello World!

```
Fred is a person at 0,0.
Make Fred say "Hello World".
```

Hello World!

```
Fred is a person at 0,0.
Make Fred say "Hello World".
```

Buzzwords!

Experimental

Natural

Educational

Object-oriented

2D animations

Language Intro

Class Definition

```
A person is a thing.
```

Object Creation

```
Alex is a person at 0,0.
```

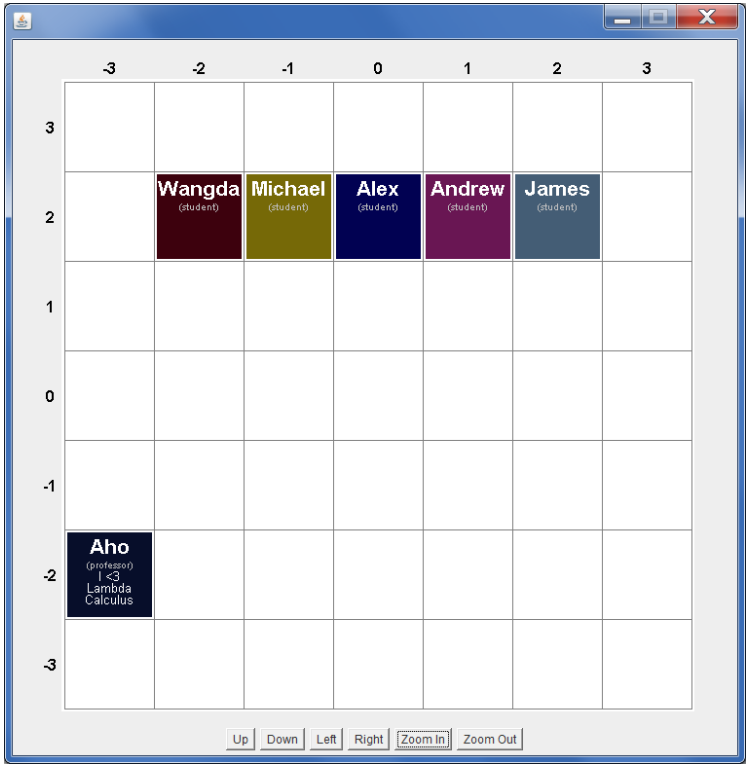
Properties

```
Alex's height is 74.
```

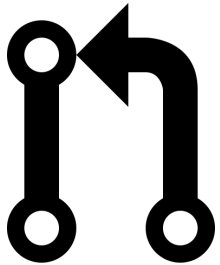
Actions

```
Make Alex move up 2.
Make Alex say "Hi".
```

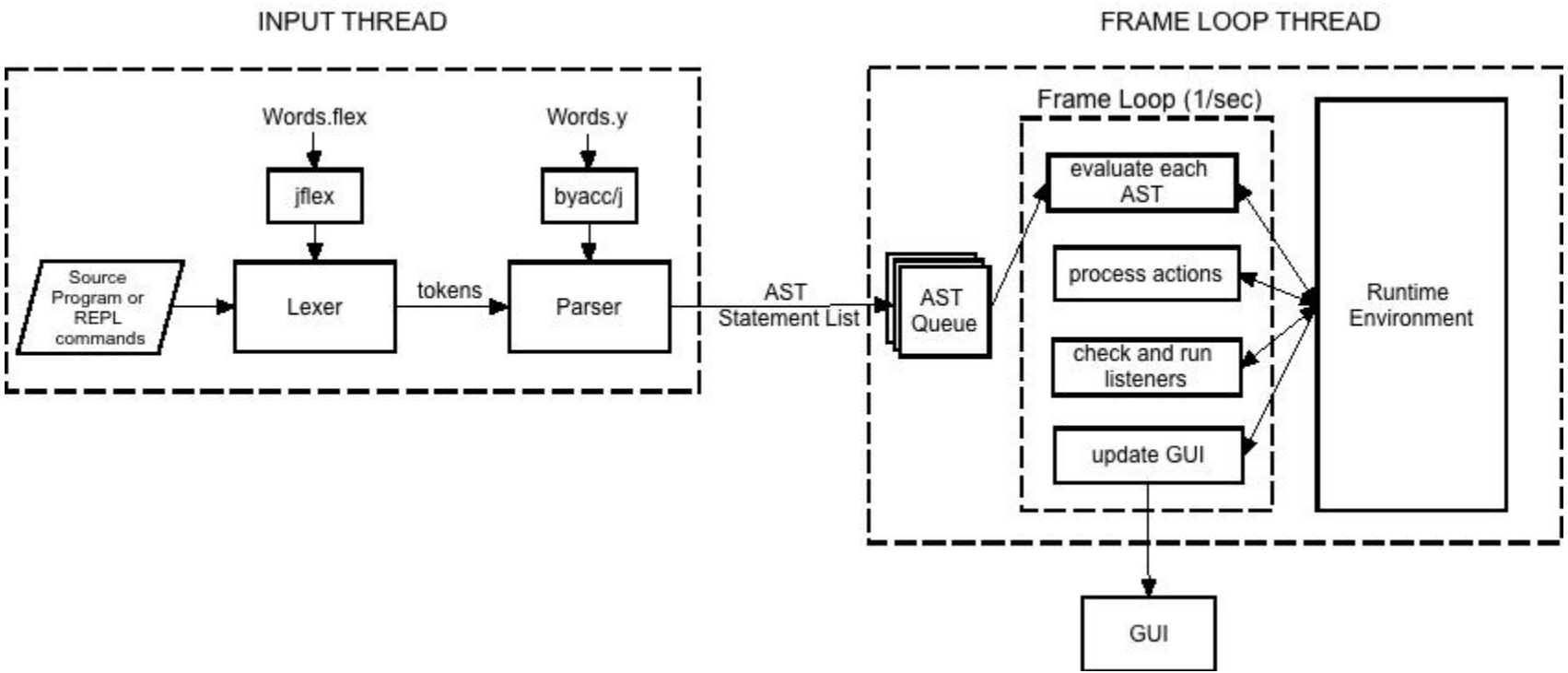
Quick Demo


-3
-2
-1
0
1
2
3
3
2
1
0
-1
-2
-3
Wangda
(student)
Michael
(student)
Alex
(student)
Andrew
(student)
James
(student)
Aho
(professor)
I <3
Lambda
Calculus
Up
Down
Left
Right
Zoom In
Zoom Out

Project Management



Under the Hood



What's Happening Here?

```
Alex is a thing at 0,0.
Bob is a thing at -1,4.

Make Bob move down 8.

Repeat 8 times {
    Make Alex say Bob's row.
}
```

What's Happening Here?

```
Fred is a thing at 0,0.
Make Fred wait 4.
Repeat 3 times {
    Star is a thing at 0,0.
    Make Star move anywhere 4.
    Make Fred say "A star at:\n" +
        Star's column + "\n" + Star's row.
}
```

Unit/Integration Testing

JUnit

Every method, every exception, every programmer

Try to think of and cover edge cases:

```
Bob's friend is Bob.
Remove Bob's friend.
```

System Testing

Test end-to-end every language feature...

...but how can we test visual output?

Logs!

Easy to diff

Fast to run

Captures errors too

```
frame #: 1
(-1,-1):
<person> Bob
(-2,-2):
<person> Alison
(0,0):
<man> Charlie
(2,2):
<child> Emily
(3,3):
<boy> Frank
```

```
frame #: 2
(-1,-1):
<person> Bob
(-2,-2):
<person> Alison
(0,0):
<man> Charlie
(2,2):
<child> Emily
"3.400000"
(3,3):
<boy> Frank
"3.400000"
```

```
frame #: 3
(-1,-1):
<person> Bob
(-2,-2):
<person> Alison
(0,0):
<man> Charlie
(2,2):
<child> Emily
"3.400000"
```

Statistics and Demo

200+ Unit tests

3,000+ lines of testing code

15,000+ lines of logs

Simply run **`ant test`**

```
test-unit:
    [mkdir] Created dir: C:\workspace\Words\report
    [junit] Running words.test.TestASTValue
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped:
0, Time elapsed: 0.033 sec
    [junit] Running words.test.TestEnvironment
    [junit] Tests run: 15, Failures: 0, Errors: 0, Skipped:
0, Time elapsed: 0.061 sec
    [junit] Running words.test.TestINode
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped:
0, Time elapsed: 0.028 sec
    [junit] Running words.test.TestINodeAdd
    [junit] Tests run: 7, Failures: 0, Errors: 0, Skipped:
0, Time elapsed: 0.05 sec
    [junit] Running words.test.TestINodeAlias
    [junit] Tests run: 3, Failures: 0, Errors: 0, Skipped:
0, Time elapsed: 0.034 sec
    [junit] Running words.test.TestINodeAnd
    ...
test-system:
    [apply] [System Test] ArithmeticTest.words OK
    [apply] [System Test] BasicTest.words OK
    [apply] [System Test] BooleanListenerTest.words OK
    [apply] [System Test] BooleanTest.words OK
    [apply] [System Test] CreateClassTest.words OK
    [apply] [System Test] CustomActionTest.words OK
    ...
```

Marco Polo with Bob & Alice

```
Whenever Bob says "Marco" {
    Stop Alice.
    Make Alice say "Polo".
    Make Alice say "".
    Repeat 3 times {
        Make Alice move anywhere 2.
    }
}
```

```
Whenever Alice says "Polo" {
    Stop Bob.
    Bob's xMove is Alice's column -
        Bob's column.
    Bob's yMove is Alice's row -
        Bob's row.
    Make Bob move right Bob's xMove.
    Make Bob move up Bob's yMove.
    Make Bob say "Marco".
    Make Bob say "".
}
```

Lessons Learned

Write good tests

Don’t be afraid to change code/architecture you’ve already developed

Seriously, write good tests

Be ambitious, reasonable, and steady

Don’t be too clever

One More Demo