GEO: Shapes that Matter

THE GEO TEAM
The GEO Team

Ruibo Li
Project Manager

Yi Guo
Language Guru

Nicolas Mesa
System Architect

Di Ruan
System Integrator

Yin Zhao
QA Engineer
Introduction
Hello World

`printf`

`printl`
if (you like GEO)

```java
string
if
    printl
else
    printl
end
```
But, for how long?

```
int while printl
end
```

I THINK I CAN.
I THINK I CAN.
Give me a shape
It’s MAGIC!

```
int int
if
  el
end
end
```
Hello World, Again!

1. Create a .geo file

```
Window window := createWindow("Hello World!", 200, 100)
```

2. Issue the command

```
yiguo@Yi-MacBook-Pro:~/Documents/GEO/Compiler$ geoc hello.geo hello
compiling hello.geo
yiguo@Yi-MacBook-Pro:~/Documents/GEO/Compiler$ ./hello
```

```
Environment

- Development Environment

- Runtime Environment
Compiler Architecture

Diagram:
- Lexer: Reading .geo file (Source code)
- Parser: Insert / Retrieve variables, Push and Pop scopes
- Code generator: Output Python program
- Functions: Add / Retrieve user defined functions
- ScopeStack: Used alongside Functions

Flow:
1. Lexer takes the .geo file as input.
2. Parser processes the file and interacts with Functions and ScopeStack.
3. Code generator produces the final Python program output.
Scope Stack

```
int b := 10

int b := a

b := a

b
```

```
b_1 = 10

b_2 = a_1

b_1

b_1 = a_1

b_1
```
Test
Project Management

- GitHub
- PyCharm
- Google Drive
- Google Docs
- Google Sheets
- Lion Mail
- Agile methodology
- Implemented feedback
- Iteration planning
- Implementation
What have we learned?
Find Difference
Open Window
Tic Tac Toe

Player 2 wins!