

GOLD



Game Oriented Language for DnD



Team members

Ezekiel Reyna
Manager

Aidan Rivera
System Architect

Timothy Chung
Language Guru

Dennis Guzman
Tester

Table of Contents

Introduction & Background

Project Timeline

Development Environment

Syntax & Usage Architecture

Testing

Demo

Inspiration

GOLD is a language inspired by Go and C and uses basic primitives to make Game development easier.

Simple functions and basic primitives

Expressive control flows

Explicit typing

Features

Safety

- Gold is statically typed

Familiarity:

- Similar syntactically to Go and C and can be picked up easily by developer of any level

Development Environment



Syntax

<Operator>
[]
-, @
!, -(negation)
+, -(subtraction)
*, /, %
<, >, <=, >=
==, !=
&&,
=

<Built-in types>
Bool // bool my_bool = true
Int // int my_int = 100
Float // float num = 12.89
String // string name = "hello"
Array // int[3] b = {0,0,1};
Pointer // string test = "Hey"; // string ~ptr = @test;

Syntax

<Function declaration>

```
func sayHello(string name) void {
    string helloStr = sprintf("Hello
%s", name);
    println(helloStr);
    return;
}

func main() int {
    sayHello();
    return 0;
}
```

<Control flow>

```
func main() int {
    int x = 5;

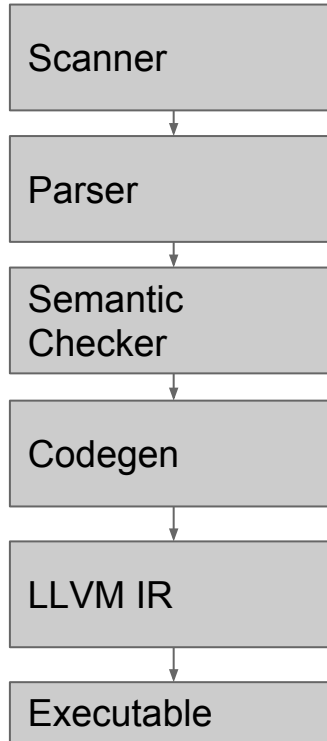
    if (x > 2) {
        print("foo\n");
    } else {
        print("bar\n");
    }

    return 0;
}
```


Syntax

<Built-in functions>
print // print string
println // print with newline
sprint // format and return a string
input // takes an input
atoi // string to int
rand // random number
srand // seed random number generator

Architecture



Test suite

`./run.sh -a`

To easily run unit test

`./run.sh func-simple-args`

To run a single test program

`./run.sh -l`

To show all available test/example programs

```
Cleaned!
dennisgzmn@dyn-209-2-219-133:~/Desktop/plt/final-proj$ ./run.sh -a
Built!
##### fail-array #####
FAILED
##### fail-assign1 #####
FAILED
##### fail-assign2 #####
FAILED
##### fail-assign3 #####
FAILED
##### fail-assign4 #####
FAILED
##### fail-for1 #####
FAILED
##### fail-main #####
FAILED
##### fail-scoping-simple-blocks #####
FAILED
##### fail-string-negation #####
FAILED
##### fail-undeclared1 #####
FAILED
##### succeed-alphanumeric-names #####
SUCCESS
##### succeed-array-element-change #####
SUCCESS
##### succeed-array-with-len-int #####
SUCCESS
##### succeed-assign-int #####
SUCCESS
##### succeed-assign-string #####
SUCCESS
##### succeed-atoi-happy #####
SUCCESS
##### succeed-basic-binops #####
SUCCESS
##### succeed-check-type-assign #####
SUCCESS
##### succeed-comparison-int #####
SUCCESS
```

Demo
