Sprint 4 Plan

Go Compiler

April 25 - May 2

Kyle Remmert (Scrum Master), Petar Zaninovich, Trevor Ching, Vincent Kim (Product Owner)

Task listing, organized by user story

Sprint 4

(17)User Story 1: As a user, I would like lexical scoping so my variables are unique across scopes.

Assigned: Kyle

Tasks:

- **(4)Kyle:** Add unit tests to account for all scoping problems
  - Global variables remain the same in recursion/ across multiple functions
  - Using the same parameter inside recursion
  - Variables with same name across different scopes
  - Check if you can't see variables outside for loops and if statements
- **(13)Kyle:** Function scoping for calling a function that is not yet created in the IR
  - Change how scoping works for forward declared functions so these functions have their own scope
  - Arrange function scoping
  - Test that functions work correctly

(33)User Story 2: As a user, I would like my Go program to typecheck correctly to keep language consistency and provide with meaningful error messages.

Assigned: Petar

Tasks:

- (10)**Petar:** Multiple assignments: check the type of every variable before assigning the values
- (8)**Petar:** Function calls are type checked in the parameters with the function being called.
- (8)**Petar:** Return types and length match function signature checked inside a function
- (7)**Petar:** Non primitive types check before accessing and writing to the type

(19)User Story 3: As a user, I would like to be able to use a map

Assigned: Vince

Tasks:

- **(1)Vince:** Implement Map type IR Nodes from the GoLang ast.

- **(3)Vince:** Create the Map type node with a Hashmap accessed similar to Struct nodes
- **(8)Vince:** Allow composite lits to be constructed for a Map type node
- **(2)Vince:** Read and writing to a map using the index node
- **(5)Vince:** Add read and writes specialization inside the frameslot write nodes and the corresponding read/writes for Map nodes.

(15)User Story 4: As a user, I would like the append builtin

Assigned: Trevor

Tasks:

- **(1)Trevor:** Create a new Rootnode for the append builtin in GoContext, then add the function to the function registry.
- **(1)Trevor:** Create Append file which will read in the frame arguments and append it to a slice.
- **(3)Trevor:** Allow a whole slice to be passed in with multiple single objects to be appended at the end of the slice.
- **(5)Trevor:** Allow a slice to be appended to another slice.
- **(5)Trevor:** Allow a portion of a slice to be passed in and append another slice on the end of it.

(14)User Story 5: As a user, I would like to be use fmt println.

Assigned: Trevor

Tasks:

- **(1)Trevor:** Add the fmt println root node to the function registry.
- **(2)Trevor:** Redo the fmt println file to handle taking in variable amounts of arguments.
- **(3)Trevor:** Loop through the arguments and append it to a string builder, then print the result.
- **(5)Trevor:** Fix selector expressions to handle both imports and structs when selecting a method or a field.