Sprint 1 Release 2 Report

Go Compiler

April 11th, 2018

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

Actions to Stop Doing:

Lateness to meetings is not tolerated.

Actions to Start Doing:

As a team, we need to group up and show each other our code when merging. It will help our understanding of the structure of the code and help keep each other updated.

Actions to Keep Doing:

We plan to keep weekly meetings. It makes us more productive, lets us redirect focus quickly and keeps us updated with each others progress.

Work Completed/Not Completed:

Sprint 1

(21)User Story 1: As a user, I would like to use pointers in my Go source code.

Assigned: Trevor

Tasks:

- **(2)Trevor:** Add the missing Golang ast tree nodes into the parser and collect the child nodes involved.
- **(5)Trevor:** Figure out a way to either model the memory for Go or mimic memory without implementing the Unsafe api
- **(5)Trevor:** Star nodes execute on the expression and returns the value pointed in the frameslot
- **(3)Trevor:** Unary & nodes return the frameslot of the variable
- **(6)Trevor:** Dereferencing pointers can be written into and read from using an index expression

(8)User Story 2: As a user, I would like to be able to import a library so I could use more functions.

Assigned: Kyle

Tasks:

- ✓ **(1)Kyle:** Create the nodes in the IR for import
- ✓ **(3)Kyle:** Create the SelectorExpr node in Truffle
- ✓ **(3)Kyle:** Update invoke to allow for SelectorExpr or Ident nodes
- ✓ **(1)Kyle:** Install the package if it is not installed.

(5)User Story 3: As a user, I would like to be able to call a function.

Assigned: Trevor

Tasks:

- ✓ **(2)Trevor:** Call a function without parameters
- ✓ **(2)Trevor:**Create/Declare a function with 0 parameters
- ✓ **(1)Trevor:** Setup the IR for a fieldlist

(5)User Story 4: As a user, I would like to be able to use all of Go's types.

Assigned: Vince

Tasks:

- ✓ **(1)Vince:** Add float(32), double(64) type nodes to Truffle
- ✓ **(1)Vince:** Add float, double type IR nodes
- ✓ **(2)Vince:** Create visit methods for each type of node
- ✓ **(1)Vince:** BasicLit GoIRFloatNode, GoIRCharNode accounted for in the visit methods

(7)User Story 5: As a user, I would like functions to be able to return a value

Assigned: Petar

Tasks:

- ✓ **(4)Petar:** Add the corresponding IR nodes to the IR tree and switch statement object factory.
  - Field, fieldlist, and return.
  - Add associated ir to truffle node transformer methods.
- ✓ **(2)Petar:** Create Truffle node return stmt
  - On execute this will throw the value to where it was called.
- ✓ **(1)Petar:** Assign variables with function return value

(13)User Story 6: As a developer, I would like Arrays and Slices to work in a consistent manner. Arrays and Slices do not work in the same fashion when being executed during reading and writing.

Assigned: Trevor

- ✓ **(1)Trevor:** Create super class type for arrays and slices, will be able to read, insert, len, cap.
- ✓ **(3)Trevor:** Create new write ir to truffle node transformer to distinguish between a read and a write to an index node.
- ✓ **(1)Trevor:** Figure out what should be returned when a array type or a slice type is executed.

- ✓ **(3)Trevor:** Calculate the offset for slices when reading and writing to a slice.
- ✓ **(5)Trevor:** Handle slice operators on arrays and slices, returning a new slice.

The team managed to complete all of the user stories associated with this sprint.