

Sprint 2 Plan

Go Compiler

April 11 - April 18

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

Task listing, organized by user story

Sprint 2

(21)User Story 1: As a user, I would like to call and create functions with parameters

Assigned: Kyle, Trevor

Tasks:

- **(2)Kyle:** Read in the Functype nodes and Fieldlist nodes from the AST file
- **(10)Kyle:** Add the field nodes into the function frame descriptor and the lexical scope
- **(7)Trevor:** Create write nodes out of each field listed so the invoke will recognize the arguments.
- **(3)Kyle:** Execute the write nodes at the beginning of the function

(21)User Story 2: As a user, I would like to be able to use a struct and the fields within a struct

Assigned: Vince, Trevor

Tasks:

- **(5)Trevor:**Change the call target of the program from the main function to a File node, so that execution of global variables and type declarations can be made
- **(1)Vince:**Read in the struct type node from the AST parser.
- **(5)Vince:**Create the struct type node with a symbol table.
- **(2)Trevor:**Allow composite lits to either assign variables by name or by order
- **(3)Trevor:**Selector expressions read from the frameslot and access the struct variable in its symbol table.
- **(5)Vince:**Allow structs to define themselves in the lexicalscoope at the top level so that they can add themselves as a field

(3)User Story 3: As a user, I would like to diff my program output with Go output

Assigned: Petar

Tasks:

- **(3)Petar:** Change gt - this runs our entire program sequence
 - Run and output go program to file
 - Output our program to a file
 - Diff the 2 files

(18)User Story 4: As a user, I would like my functions to be able to return multiple things

Assigned: Petar

Tasks:

- **(3)Petar:** Be able to assign the values from the multiple returns
 - Change how assignment works, in parser and in the ir to truffle transformation
- **(1)Petar:** Change the Truffle Nodes to allow the return of multiple values
 - Returnstmt node
- **(7)Petar:** Type Check the return with the assignment variable
- **(7)Petar:** Check Number assignments corresponds with number of things returned