

[2019-04-30]

- Initialized project
 - Forked and cloned mal repository
 - Cleaned up repository for my implementation

[2019-05-01]

- Step 0: The REPL
 - Created step0_repl.clj file
 - Added stubs for read, eval, print, and rep functions that just thread user input

[2019-05-02]

- Step 0: The REPL
 - Added an input loop with a user prompt

[2019-05-03]

- Step 1: Read and Print
 - Implemented tokenizer, form reader, and basic atom reader functions
 - Began implementing list reader but it has bugs
 - Created printer.clj file
 - Created form printer function
 - Read-Eval-Print loop uses new reader and printer functions

[2019-05-04]

- Step 1: Read and Print
 - Finished implementing bug-free list reader

[2019-05-07]

- Step 1: Read and Print
 - Added nil, true, false, and strings
 - Added error checking for paired punctuation
 - Added keywords, vectors, and maps
 - Added comments

[2019-05-08]

- Step 2: Eval
 - Fixed issue where mal lists were not being stored as Clojure lists
 - Added REPL environment to contain bindings for functions and values
 - Added eval-ast function to evaluate code
 - Updated EVAL function to actually evaluate an AST with an environment
 - Updated eval-ast function to evaluate vectors and maps
 - Map keys can only be strings or keywords
- Step 3: Environments
 - Added environment objects with support for nesting

- Added ability to add bindings and retrieve bindings from an environment
- Added def! special form for permanent bindings
- Added let* special form for temporary scoped bindings
- Step 4: If Fn Do
 - Added support for inserting bindings on environment creation
 - Updated printer for printing function values
 - Added do special form for evaluating multiple forms
 - Added if special form for selectively evaluating forms based on a conditional form
 - Added fn* special form for creating functions

[2019-05-09]

- Step 4: If Fn Do
 - Added core library of functions
 - Added support for variadic function parameters
 - Added not function defined within mal
 - Added many string functions to core

[2019-05-10]

- Step 5: Tail Call Optimization
 - Added tail call optimization to prevent stack overflow in recursive functions when possible

[2019-05-15]

- Step 5: Tail Call Optimization
 - Fixed two issues with TCO and thus finalized it

[2019-05-16]

- Step 1: Read and Print
 - Fixed failing tests regarding metadata, deref, and bad string form
- Step 3: Environments
 - Fixed failing test regarding unbound symbols
- Step 4: If Fn Do
 - Fixed failing test regarding string quoting and escaping

[2019-05-19]

- Step 5: Tail Call Optimization
 - Now capturing all Throwables so Errors don't cause program exit
 - Fixed blank/comment-only lines not working
- Step 6: Files, Mutation, and Evil
 - Added read-string function to turn strings into mal data structures
 - Added eval function to evaluate mal data structures
 - Added slurp function to return file contents as a string
 - Added load-file function to evaluate mal file

- Added support for atoms
- Added support for running mal files from the command line
- Added support for command line arguments to mal programs
- Step 7: Quoting
 - Added support for quoting
 - Added support for quasiquoting, including unquoting and unquote splicing
 - Added support for vector quoting

[2019-05-20]

- Step 8: Macros
 - Added support for defining macros
 - Added support for macro expansion
 - Added nth, first, rest functions
 - Added cond and or macros

[2019-05-27]

- Step 9: Try
 - Added support for try*/catch* forms with exception handling
 - Added throw function
 - Added apply function
 - Added map function
 - Added nil?, true?, and false? functions
 - Added symbol and symbol? functions
 - Added keyword and keyword? functions
 - Added vector and vector? functions
 - Added sequential? function
 - Added hash-map and map? functions
 - Added assoc, dissoc, get, and contains? functions
 - Added keys and vals functions

[2019-05-30]

- Step A: Mal
 - Added support for metadata
 - Added *host-language* symbol and message
 - Added readline function
 - Added time-ms function
 - Added conj function
 - Added string?, number?, fn?, macro? Functions
 - Added seq function
 - Added gensym to prevent symbol shadowing