Milestone #1 – GOCaml

<u>Team strategy</u>

- Pair programming at all time

- Work was done in a separate repository and only the latest commit will be pushed in Sable's repository

<u>Technology used</u> Menhir & Ocamllex

<u>Workflow</u> The currently submitted milestone does not contain all desired features. A huge workload from other classes (namely COMP417 and COMP527) prevented both teamates to devote a sufficient amount of time on this milestone, and that despite a 27 hour rush on the last day, which allowed the team members to improve their skills with Menhir and Ocamllex as well as test their aptitude in situation of prolonged sleep deprivation.

<u>Lexer</u>

- *Semicolon insertion*: after attempting to directly insert a SEMICOLON token in the token stream, it was found that it was easier to raise a flag to control what token a linebreak returns. Thus whenever a token which can be followed by an option semicolon is encountered, a flag is raised to return a SEMICOLON token if the next token is a linebreak.

- The rest of the lexer was built in a conventional fashion.

<u>Parser</u>

- CONTINUE and BREAK are accepted outside of loops by the parser. This will later be forbidden by a traversal of the AST, by a weeder.

- Likewise the parser allows nested function declaration. This will be later forbidden by the weeder.

- The parser compiles with no conflicts.

<u>Make file</u>

- No make file is currently available, so <u>we are sorry to announce that the compiler cannot be used yet</u>.

- Although, you can comfort yourself with the make.sh file which fails to build the compiler but has nice color code to alleviate the failure.

<u>AST</u>

- The AST builder is incomplete. The partial work can be seen in src/ast.ml. No code links the AST builder to the parser yet.

<u>Pretty printer</u>

- The pretty printer is incomplete. Naturally, without an AST builder it could not be fully developped. Partial code can still be seen in pretty.ml

Sources

- Little documentation exists for Ocamllex and Menhir, thus Real World Ocaml (https://realworldocaml.org/v1/en/html/parsing-with-ocamllex-and-menhir.html) is almost all the sources which were used.

- A line counter function was also used from a fellow programmer in StackOverflow.