## asd

## Team dynamic

We did not assign task explicitly to anyone. However, we are always keeping a backlog of task remaining and the dependencies between the tasks. Therefore, when a person has time, he can simply pick a task that does not have any open dependencies and work on it. We also try to have very descriptive git commit messages so that every team member is always aware of the current state of the project.

## Compiled Language

## We chose to compile GoLite into C++ for multiple reasons. First, it's a fairly low-level language which means the performance should be better compared to higher level languages. Second, it is a very mature language that is widely used. Third, C++ offer more functionalities that C does not, some of which makes the final code generation phase much easier.

## Syntactic checks deferred to the weeding phase

**Short variable declaration**

For short variable declarations, we use an expression list to represent the the lvalue which is supposed to be a list of identifier instead of an id list.

**Matching number of lval and rval in an assignment**

In an assignment, whether it's part of a declaration or a statement, we do not check if the number of lvalue and rvalue match.

**Verify whether break and continue are inside loops**

We simply allow break and continue wherever a statement is allowed, and check for its validity in the weeding phase.

**Allow all type expression to be used when a type is required**

For the moment, a type expression is both used to declaring a type and representing a type. This means that it contain

**Verify left part of assign is a valid lvalue**

**Check valid for arguments**

For now every part is option independently