Development Environment and Runtime

This document outlines the tools used for development and runtime of the YARTBML language.

IDE

The IDE we will be using is Visual Studio Code. Visual Studio Code integrates our code editor, compiler, and version control system all in one place for easy builds and control of our language.

Version Control System

We will be using Git as our version control system, and GitHub to host our repository in the cloud.

Testing

For testing, we will be using Go's built-in testing package.

Documentation Tools

We have our own custom in-house tool to generate our documentation into PDFs. We built this by using a GitHub Action which kicks off a workflow whenever new changes are pushed into main branch within the `docs` folder. Each document is written using Markdown and then transformed using the Eisvogel Template with Pandoc, inside a docker container.

Runtime

The YARTBML language uses Go's Garbage collector in order to free up any resources, which need to be freed up, and this is statically linked to the project when it is compiled down to an executable. As Go compiles down to binary, our interpreter essentially is a binary executable, where the runtime is dependent on the Operating System. However, all of the features and functionality of this application rely on Go's built-in data types and methods.