To: CS4500 Course Staff
From: Julian Hirn and Joshua Rosenberg
Subject: Assessment of TypeScript for future projects
Date: 09/24/20

Our overall assessment of TypeScript as a language for our future project is favorable. From a purely syntactic perspective, the language is easy to understand and follow based on how similar it is to C-like languages. We also found that flexibility provided by the language's type system made it simple to interface with some third party code. The caveat to this is that since the ecosystem is JS first, not all packages have bindings that provide types. In those cases, the upside from using TypeScript is partially lost since the compiler does not have complete type information. However, it will inference partial types based on how that package is being used, which is still better than vanilla JavaScript. Additionally, it is possible to add type definitions to untyped packages using type declaration files (.d.ts), and many untyped packages have types available via the open source definitelyTyped project.

For better or worse, there are always many different ways of doing things with TS/JS since much of the functionality of those languages are provided by these aforementioned packages. The main benefit from this is that it's easy to customize the code formatting and style. It is easy to use development packages such as prettier and eslint which take care of those things. The downside is that it's not always clear what the preferred solution for different non-trivial problems are due to the ease with which one can make use of third party packages. This uncertainty is likely to lead to some wasted engineering time.

We believe that through Node and additional packages, TypeScript has everything necessary to complete related to the command line, JSON, GUIs and networking.