

## **GatorGrader**

Implemented by Allegheny professor and students and designed for use with [GitHub](#), [GitHub Classroom](#), [Travis CI](#), and [Gradle](#), GatorGrader is an automated assessment tool that checks the work of programmers and writers. While other tools already exist to, for instance, enforce a style guide for source code or build a program, GatorGrader focuses on automating the checks and activities that are not already nicely supported. For example, GatorGrader can check how many Git commits a student or a team performed during the completion of an assignment. Along with checking for the existence of files, the tool can also count a wide variety of entities in a project submission, including words and paragraphs in technical writing, comments in source code, and fragments in either source code or program output.

The tool originally used Regex to count paragraphs in technical writing. Although using Regex was feasible, there were some errors when catching paragraphs (there were ways to make the check fail that should be possible). As such, my team was responsible for enhancing that part of the tool. As part of the enhancement my team and I did, we used an ast, CommonMark, to implement a general purpose Markdown parser. In result of the work done by my team and I, GatorGrader now uses CommonMark to count paragraphs. We also updated the documentation of the tool to allow new users to understand what the key features are.

Throughout the process of enhancing the project, I had learned a lot of new tools and ways on working in a team. I was also able to learn to collaborate with my team. Although we didn't meet as often as we probably should have, this gave me a new perspective on how teamwork can work outside of group meetings. Because of the unorthodox meeting schedule, I learned to be able to effectively communicate over group and individual messages.