Developing a Math Tutoring System by applying Language Application design

Columbus State University

James E. ellerbee_james1@columbusstate.edu

Radhouane C. radwan_shushane@columbusstate.edu

Draft revised 21 June 2020

Abstract

There is a limitation with the ability of current math tutoring software. The ability of these systems to take input, analyze it, and produce meaningful output. This project is a collaboration between Computer Science and Mathematics departments involving pedagogy and Programming Language (PL) theory and design. The methods required in designing an interpreter for a language. This project attempts to improve the effectiveness that existing tutoring and assessment software such as Symbolab and WebAssign provides. What Symbolab provides in great ability for input, it lacks in greater decomposition. WebAssign brings methods for students to practice, assess their progression, and interactivity, however it provides generic

feedback. The implementations in this project uses PL theory with system design and aims to bridge this gap.

Keywords: Programming Language, Software, Math Education, Math Learning Tool

Introduction

In an article titled Math scores stink in America. Other countries teach it differently and see higher achievement. written for USA Today, Erin Richards writes "American students struggle in math". A phrase that a lot of instructors agree with. While she compares the US's performance with other countries, one of the quintessential reasons provided: "Classes here often focus on formulas and procedures rather than teaching students to think creatively about solving complex problems involving all sorts of mathematics". In the rest of the article, Erin provides ideas for improving student's math scores. In this paper, we provide a solution to aid students with the language of math and how to connect it with numbers.

Harry Cheng in his article Teaching Math with Computer Programming can help Narrow Achievement Gap, written for EdSource, states that teaching math with computer programming can help students realize context and relevance.

Unfortunately, many people do not like math or have had bad experiences with taking the subject (Furner 2019) and In addition to these articles, we reached out to math instructors from Columbus State University using an online form to rank statements critically affecting students' ability to learn and apply math concepts. The statements that ranked the highest are: "Students must practice" and "Students must understand reason instead of relying as much on memorization". Based on this preliminary data, we can focus on the core features of the tool. In the future, we would like to include more instructors in this type of feedback.

Our language application utilizes Java as the backend for the interpreter. This grants platform independence, reliability, and the option to deploy the project to a wide range of end users. Open source Java libraries used on this project are ANTLR4 (Parr 2016), GRAL, and

JLatexMath. ANTLR provides an interface for generating an executable interpreter. GRAL is a graphing library for providing visual graphs for math functions. JLatexMath renders Latex formulas to make math look more natural in our application.

Our language application connects mathematical arguments with expressions in an effort to be a resource for students. The design of our software aims to address the aforementioned factors affecting students' ability to learn. By designing a system capable of mathematical expression evaluation and syntax written similarly to phrases seen in mathematical arguments and word problems.

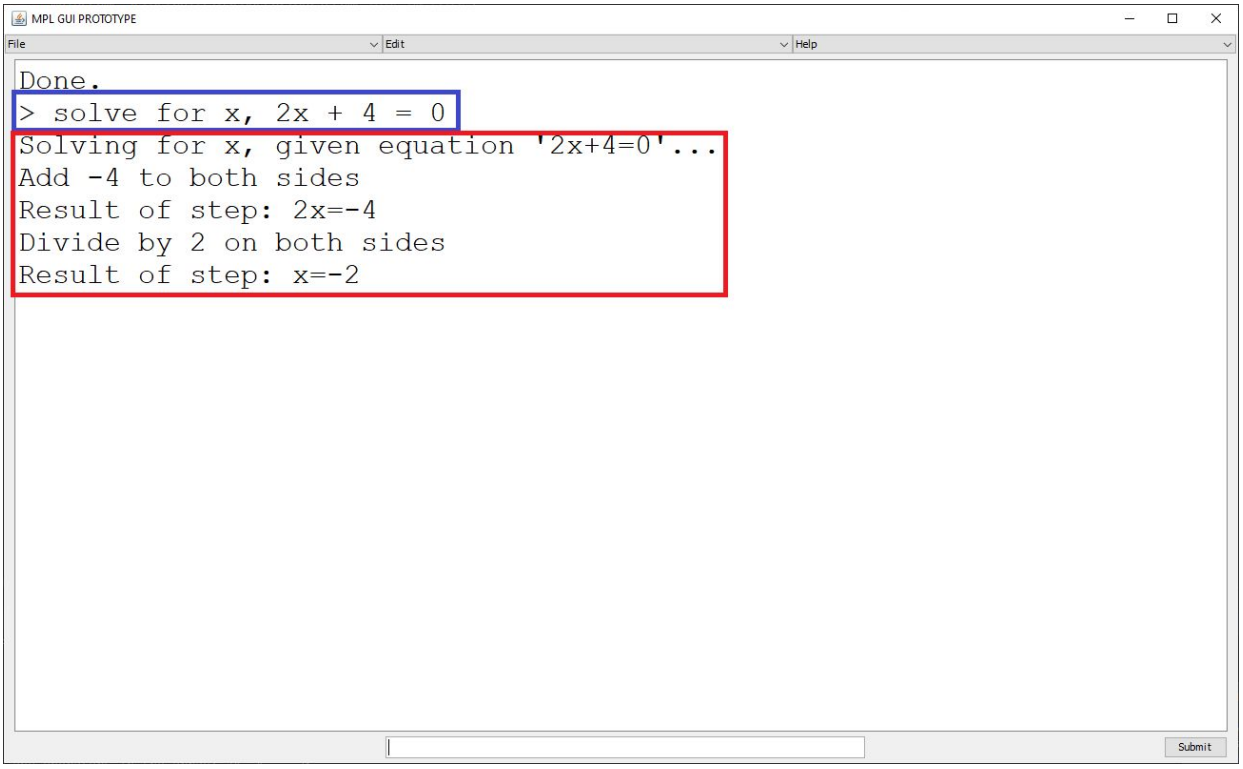

Figure 1. Math tutoring system. User Input in Blue, System output in Red

The motivation for this project comes from the desire to improve student's ability to engage with math. Academic works and existing tools such as: The BrickLayer Ecosystem, a tool to aid in teaching coding; Integrating Dynamic Mathematics Software into Cooperative Learning Environments in Mathematics, this paper shows our statistics of the positive effects on the use of this type of software with students; Eqsquest's Symbolab, our tutoring system takes inspiration from this project and attempts to improve upon it.

In the rest of this paper we plan to discuss related works, the existing math tools, turning math expressions into a tree for computation, and the conclusion and directions for further work. Section 1, related works, will highlight academic papers having significant relation to our's. Section 2, existing math tools, this section will highlight existing commercial or educational tools that students might already use. Section 3, turning mathematical expressions into a tree, will highlight how our application takes mathematical input and transforms it in order to process it, it will also highlight the current design of our system. Section 4, conclusion and directions for further work, will conclude the paper and provide several directions for further work.

Related works

This section describes academic works in the same area of research. In this section, we'll describe the contributions on the paper and how it relates to our project. We can look to these related works to see how to better develop a tool that will make a difference in mathematics education.

Bricklayer. Bricklayer is a library of graphical primitives, written in SML (Winter, Love, & Corritore 2016). This program has been used in K-12 to teach coding to students. The Bricklayer program indirectly involves Math and this gets students, or anyone using the program as the authors point out that the program is for all ages and coding backgrounds, engaged in thinking about how to engineer design. This is good for circumventing attitudes or any biases toward math as learning math directly for some can be a challenge. However, this project more directly teaches students coding, but it highlights how to get students engaged. Our project does directly involve math, simple coding and language. We are careful to ensure our program engages students to enable learning as we are concerned with attitudes student's have when attempting to learn math.

Bl0ck3d. Bl0ck3d is an educational software, it attempts to mimic the use of block-based manipulatives used to teach early math students how the base ten number system works (Young & Watkins 2018). This work contributes findings in the use of software to aid students in learning math.

Existing Math tutoring tools

This section describes popular commercial tutoring tools that may students use to aid themselves in math. In this section we'll give a description of the tool, the perceived limitations, and how our program potentially solves it.

Mathway. Mathway is a commercial application that is developed by Mathway LLC that provides students with tools to solve equations, simplify expressions, and evaluate limits. The interface helps students relate to the application. It looks like you're texting a person who is perfect at doing and explaining math. This kind of interface is powerful because it provides a familiar environment for a potentially unfamiliar problem.

Symbolab. Symbol is a commercial application that is developed by Eqsquest. Symbolab appears to be a language application that allows students to give it commands to solve their problem from a wide number of math topics. The application is powerful because

In addition to taking command from the student, it also allows for formatting of math directly in the input field.

Webassign. Webassign is a commercial application developed by Cengage for instructions to assign work to their students. Webassign has continuously released improvements to their interface. Their error reporting is able to give hints for a problem depending on context, but we think that this can be improved further by decomposing the problem using our language application.

Turning mathematical Expressions into a Tree,

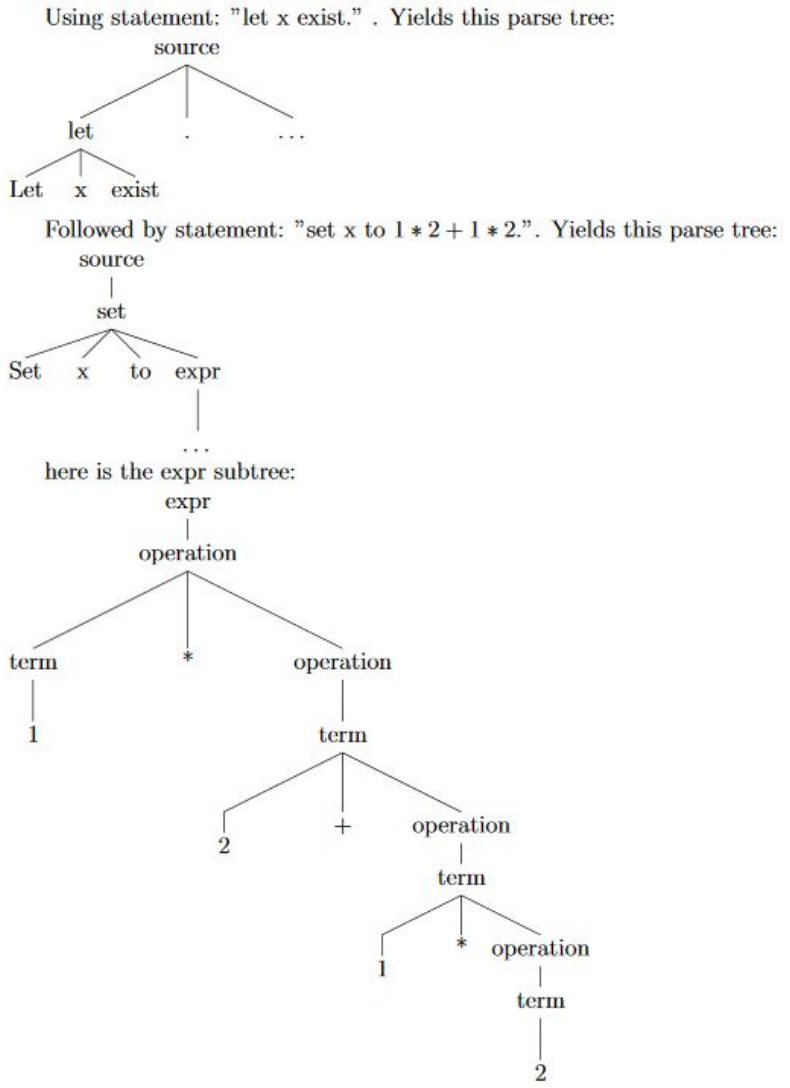

Figure 2. Parse Tree for input

We use a tree structure to give context to the statements, or instructions, in our language. We can then walk this tree in order to perform functions depending on that context. We wanted

this math language to incorporate basic programming instructions, such as storage and retrieval of values stored in variables, as well as support for accurate mathematical expressions. Then building the tree, our program respects order of operations as seen in figure 2. A limitation of this is mutually left recursion (Moore 2000), rules that derive into themselves and to infinity. We applied various techniques that do not incorporate left recursion, reducing ambiguity.

Designing an user interface. It is said that we use tools because they are convenient. We have been working on a graphical user interface for our tutoring system which will eventually be extended with our scripting language. Our math tutoring system incorporates patterns to execute implemented commands inputted through a text input field.

Math students value immediate feedback when learning and practicing math concepts.This is important in designing a clear distinction in student input and system output.As seen in Figure 1, 3, and 4. Our tutoring system's interactive mode in which the system walks the student through the problem and prompts the student for input and provides feedback for that step.

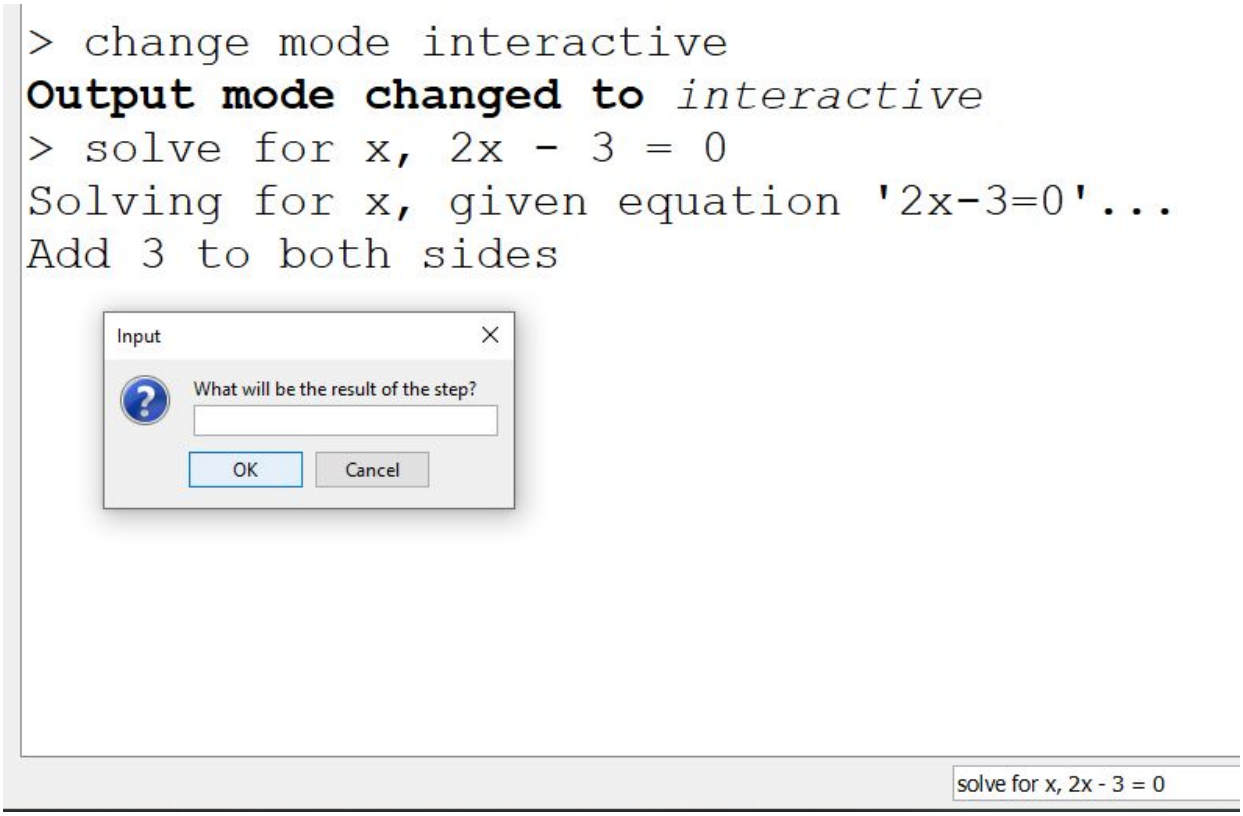

Figure 3; interactive mode and solve command

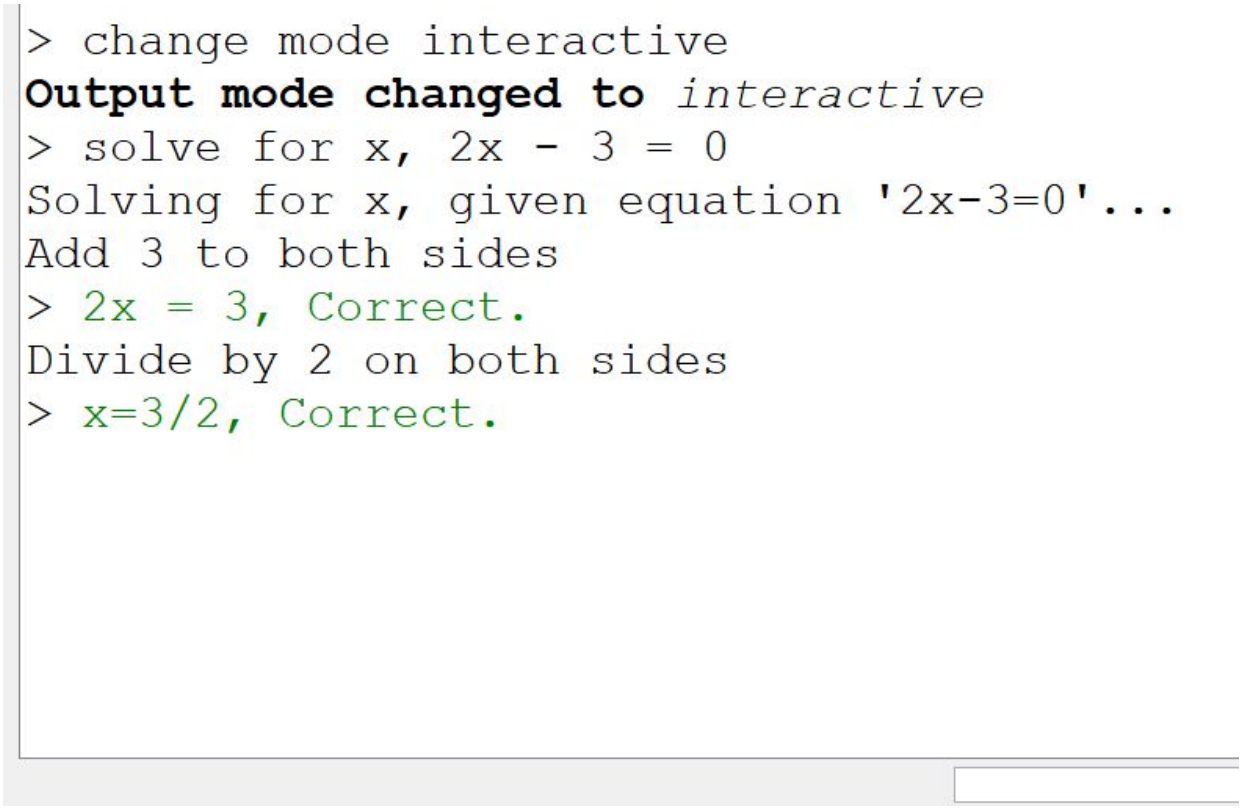

Figure 4; instant feedback in interactive mode

Other notable features. Our program implements logging in the form of a profile. When the student performs an activity the app can record relevant information about that event and record it on the user profile. The idea of this feature allows exporting of data to provide progress to an instructor in order to accomodate a student since instructors are limited in their ability to single out students based on their current understanding of math. Another prominent feature of this application is a graph plotter. Lastly, a math pad that renders math as students would naturally see, this feature is still currently currently in development before it can be included in the main tutoring application

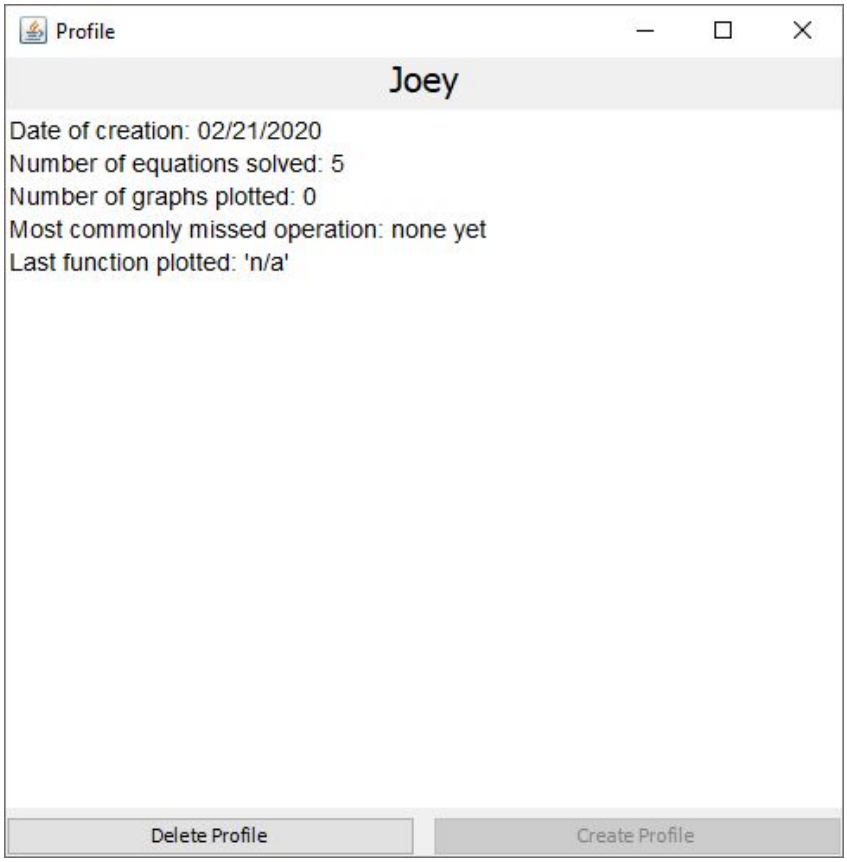

Figure 5; User profile with example name

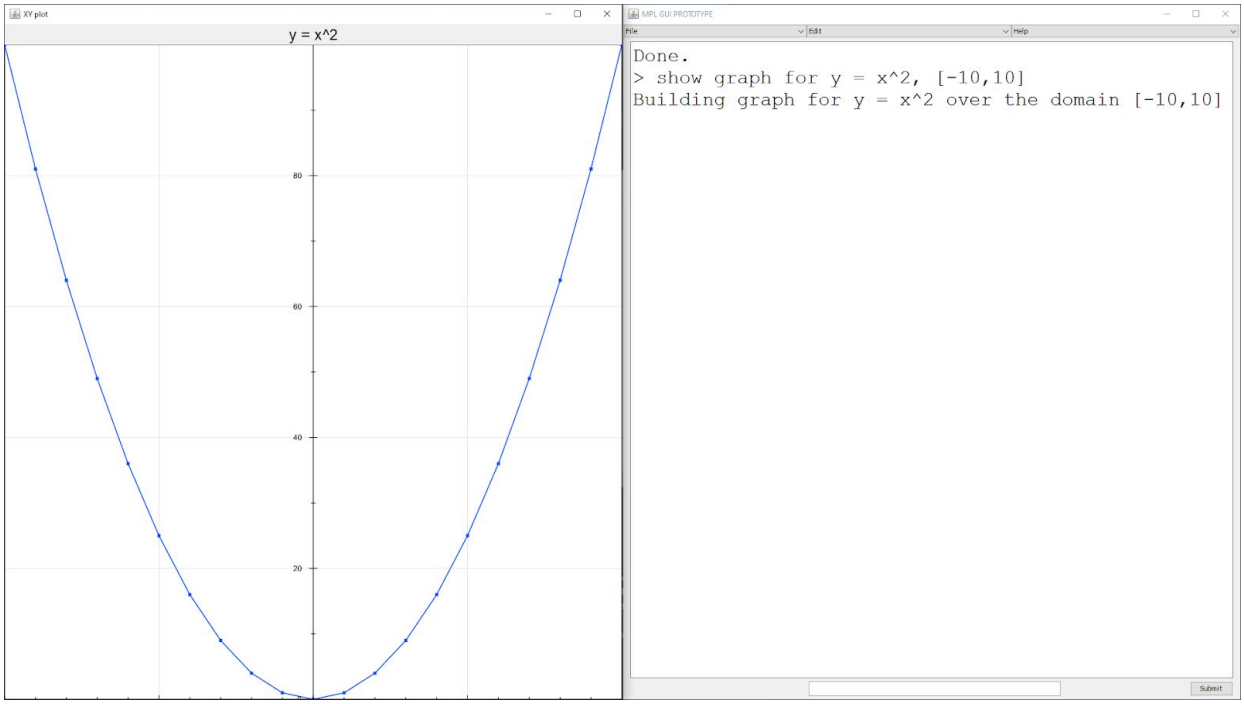

Figure 6; graph plotter

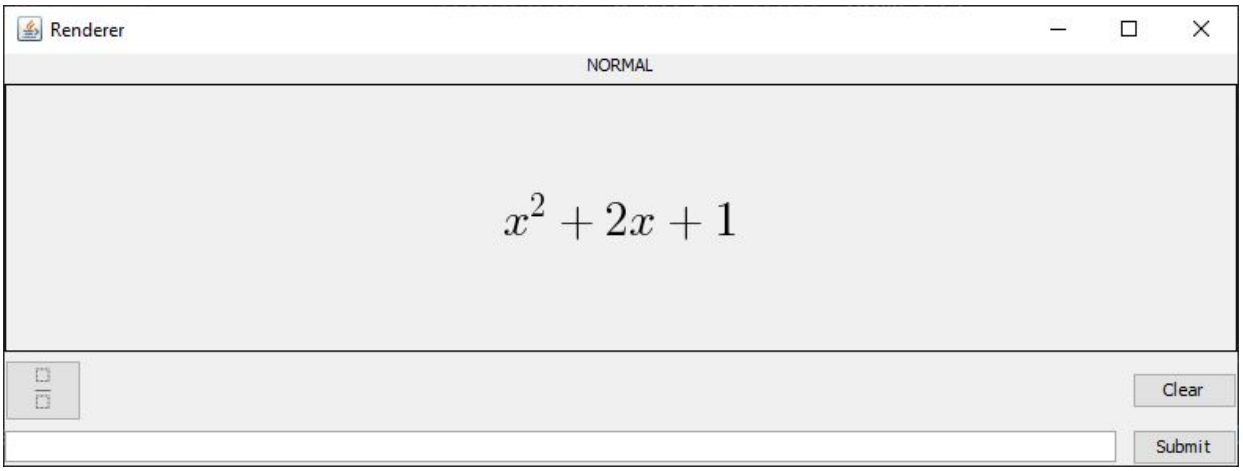

Figure 7; math pad input

Discussion

This application breaks an expression into its step by step process and uses the context to provide feedback at each step rather than just at the end, as seen in figure 4. The idea is that this is useful for the tutor and tutee as it allows the time consuming process of working through a problem to be simplified by working with the tool, freeing the tutor to focus on concepts and target problem areas. The application performs data collection, as seen in figure 5, which can be used to generate a report to pinpoint problem areas.

Conclusion and Directions for further work

Directions for further work on this application includes adding more logical control structures and improving the core math programming language. Incorporate the language into the math tutoring system in the form of a scripting mode. Perform trials of software with math

students in order to gauge usefulness, collect and analyze data on these trials. Continue to improve the application and interface based on features requested by math instructors and students. We have made an effort to include a mathpad-like feature that renders math formatted text but this needed further development and include it as the main way the math tutoring system takes student's input. Improve data from interactions with the application to generate an in depth report that can be used to aid learning.

References

Joseph M. Furner, *Math Anxiety Trends: A Poor Math Attitude Can Be a Real Disability,* Journal of Advances in Education Research, Vol. 4, No. 2, May 2019.

Terence Parr, *The Definitive ANTLR 4 Reference,* The Pragmatic Programmers, 2007

V Winter, B Love, C Corritore, *The Bricklayer Ecosystem - Art, Math, and Code*, Trends in Functional Programming in Education, 2016.

C Young, A Watkins, G Christie, *Interface design for a math manipulative application,* Journal of Computing Sciences, 2018

Robert C Moore, *Removing Left Recursion from Context-Free Grammars*, 6th Applied Natural Language Processing Conference: 249–255, 2000