

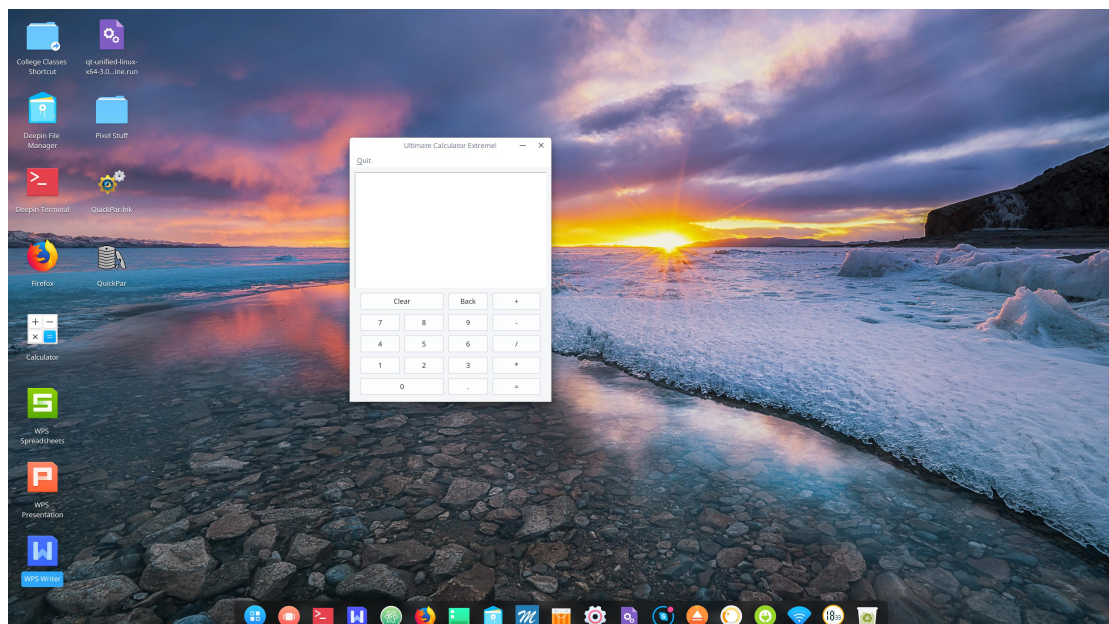
Qt Graphical Calculator Project

Code Development

The code for the calculator was written with C++ using the QT library for the graphical components of the project. The project only consisted of two classes: the Calculator class (inheriting from QMainWindow for the main project), and the Buttons class (inheriting from QFrame for the layout of all buttons). I could have put everything in one class or not used a class at all, but I wanted things to be more simple and structurally laid out, so I decided to go this method with my classes. The Buttons class has QPushButton objects for every button shown in the calculator, and they are laid out in a grid format to look like a real calculator. The Calculator class has an object of type buttons so as to use its layout in the main display of the application. Every button in the Buttons class also has its own signals it sends to the Calculator display object to update the screen when a button is pressed. The overall combination of the two classes when run with QT, runs a calculator GUI that allows for the user to enter valid expressions which will be calculated when the equals sign is pressed. The application also handles validation of operators and invalid expressions as the user is typing in information. Once an expression is entered and the equals sign is pressed, a parser then handles the computation. I pulled the parser project from GitHub, and the project was free to use. The parser is not mine, and I do not take any credit for how it accomplishes what it does. I just used it to show the answer of the expression on the screen.

Results

The application is shown below, and it runs very well. The expressions are calculated very quickly, and it is nice that it handles expression validation in all ways for this simple calculator. The calculator will keep running until the user quits the program, so as many expressions can be calculated as the user desires. QT overall is a very powerful library of applications, and it can be used to make amazing GUI projects. This helped me explore the uses of QT a little more, and I hope to keep working with it in the future.



Code Source Files

All source files for this project are uploaded to Git Classes under the link (<https://git-classes.mst.edu/2018-SP-CS1585-B/2018-sp-b-project02-djbn65>).

The code files were too long to have screenshots of here, but everything is viewable through the above link. I wrote the `buttons.h`, `buttons.cpp`, `calculator.h`, and `calculator.cpp` files. The other `.h` and `.cpp` files were for the parser, and the `.pro` file was for the QT project.