

# Type Through the Bible (C++ Edition)

## A Typing Game of Biblical Proportions

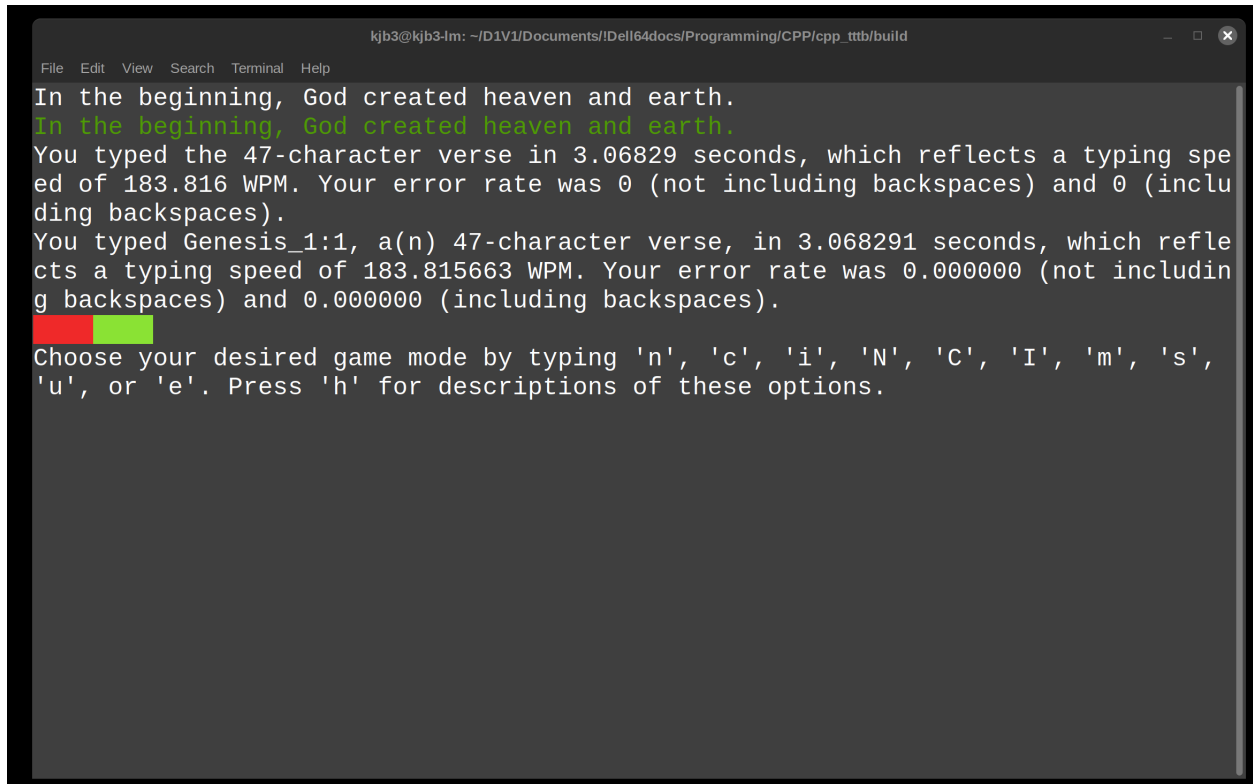
By Ken Burchfiel

Released under the MIT License

[Link to GitHub repository](#)

[Link to game download page on Itch.io](#) (password: `microseconds`)

[Note: this Readme is a very early work in progress! More information will be provided soon.]



```
kjb3@kjb3-lm: ~/D1V1/Documents/IDell64docs/Programming/CPP/cpp_ttb/build
File Edit View Search Terminal Help
In the beginning, God created heaven and earth.
In the beginning, God created heaven and earth.
You typed the 47-character verse in 3.06829 seconds, which reflects a typing speed of 183.816 WPM. Your error rate was 0 (not including backspaces) and 0 (including backspaces).
You typed Genesis_1:1, a(n) 47-character verse, in 3.068291 seconds, which reflects a typing speed of 183.815663 WPM. Your error rate was 0.000000 (not including backspaces) and 0.000000 (including backspaces).
Choose your desired game mode by typing 'n', 'c', 'i', 'N', 'C', 'I', 'm', 's', 'u', or 'e'. Press 'h' for descriptions of these options.
```

## Introduction

[To be completed soon!]

## Download instructions

I plan to make Linux, Windows, and OSX copies of TTTB available [at this page](#) on itch.io . Until the documentation for the game is more complete, the page will be password-protected; you can access it by entering the password `microseconds`.

A Linux binary is currently available, and Windows and OSX ones will be added soon.

The game is free to download, but donations (while not expected) are greatly appreciated.

## Gameplay instructions

[To be completed soon!]

## Compilation instructions

These instructions should work for Linux, OSX, and Windows. Please let me know if you encounter any issues.

1. Download [CMake](#) if you haven't already.
2. Clone the [cpp-terminal](#) and [csv-parser](#) libraries, both of which are permissively licensed, and build them using CMake. **\*\*Note:** I recommend switching the `BUILD_SHARED_LIBRARIES` option within the cpp-terminal [CMakeLists.txt](#) to OFF for all platforms in order to make your resulting code more portable.

(To build each library using CMake, (a) create a 'build' folder within the folder on your computer that contains the library; (b) navigate within this folder using your terminal; (c) run `cmake ..` in preparation for the build; and then, if that command was successful, (d) run `cmake --build .` Don't forget the space and period at the end!)

3. Copy these folders to a directory of your choice, or leave them in your downloads folder if you prefer.
4. Clone [Type Through the Bible](#) and move it to a directory of your choice.
5. **IMPORTANT:** Within TTTB's [CMakeLists.txt](#) file, you *must* replace the existing paths to my copies of the cpp-terminal and csv-parser libraries within the `add_subdirectory()` calls with paths to *your* copies of these folders (at least for the platform(s) for which you're building TTTB). Within each `add_subdirectory()` call, the first path is to the actual 3rd-party-library's location on your computer; the second path tells CMake where to place some additional files related to that library. (You don't need to create this second folder on your system; CMake will create it automatically.)
6. Create a build folder within the same folder that holds TTTB's [CMakeLists.txt](#) file, and navigate into it.
7. Use CMake to build TTTB. (See above steps for reference if needed.) Once it has been build, copy the resulting executable (e.g. `tttb` or `tttb.exe` on Windows) into your build folder. (Windows will likely place it into a separate subfolder, but it must be placed within your main build folder in order for the relative paths used by the program to work correctly.)
8. Next, you'll need to build an executable version of the `tttb_py_complement.py` file. First, download [Pyinstaller](#) if you don't have it already. Also, download Python (i.e. via [Miniforge](#) if it's not already on your system).
9. Make sure that Pandas, Numpy, and Plotly are all installed within the Python environment that you plan to use to run Pyinstaller. (These can be downloaded via conda-forge.) Otherwise, the executable version of your `tttb_py_complement.py` file won't work correctly.
10. Once you have Python and Pyinstaller set up, run `pyinstaller tttb_py_complement.py` within the 'build' folder that you created within the main TTTB folder. This will create an executable version of this file, along with an `'__internal'` folder that contains important library components, into a `'dist/tttb_py_complement'` subfolder within your 'build' folder. **\*\*Cut and paste both the `__internal` folder and the `tttb_py_complement` executable into your build folder, as that's where the program will look for them.\*\***
11. Navigate up to the main `cpp_tttb` folder (e.g. via `cd ..` in your terminal, assuming you're still in the build folder) and run the `create_release_folder.py` Python file. This will create a new copy of TTTB with blank output files rather than the existing files within the TTTB directory.
12. You should now be able to begin playing TTTB!

## Acknowledgments

I am grateful for Ronald L. Conte Jr. for his work on the [Catholic Public Domain Version](#) of the Bible (the translation that TTTB uses).

This game is dedicated to my wife, Allie. I am very grateful for her patience and understanding as I worked to put it together! My multiplayer gameplay sessions with her (yes, she was kind enough to play it with me) also helped me refine the code and improve the OSX release.

Blessed Carlo Acutis, pray for us!

[More acknowledgments to come!]