

# Type Through the Bible (C++ Edition)

## A Typing Game of Biblical Proportions

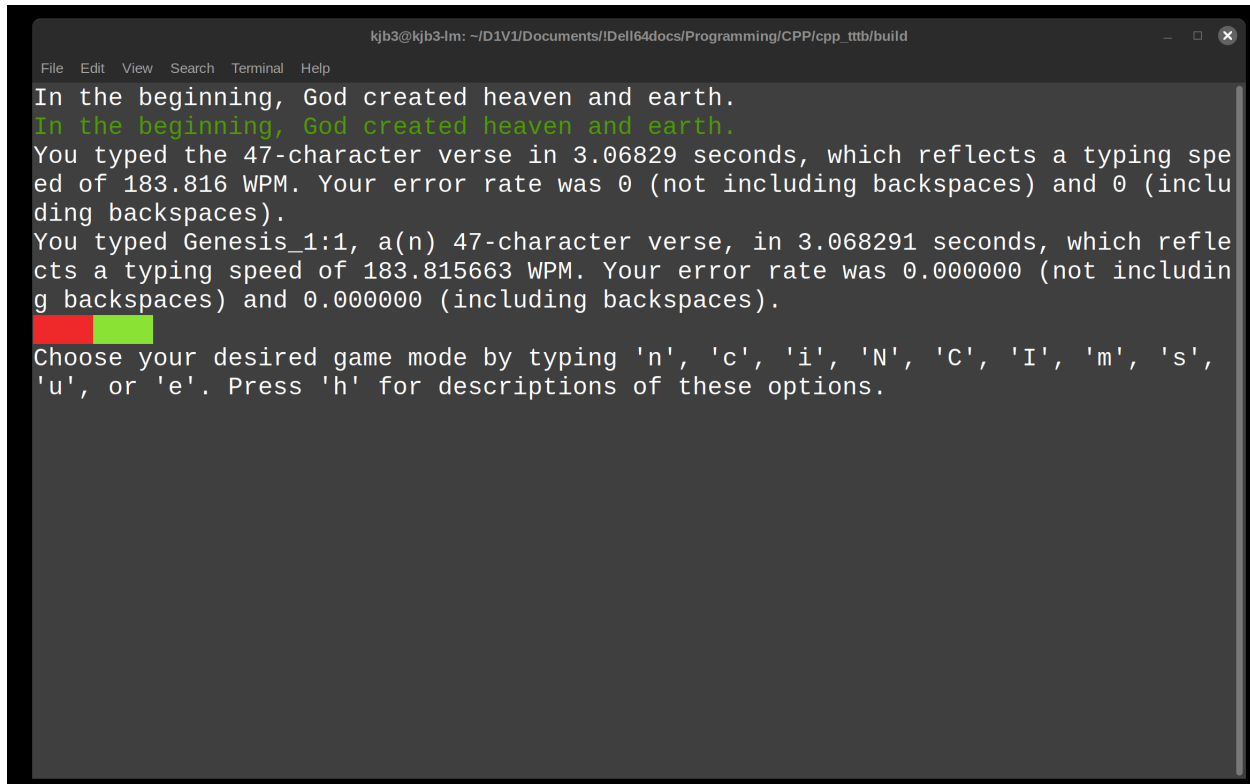
By Ken Burchfiel

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

Type Through the Bible (TTTB) allows you to practice your keyboarding skills by typing the Bible verse by verse. It contains both single-player and multiplayer modes, and also offers (via a complementary Python script) a wide variety of interactive visualizations.

## 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.

Make sure to download both the zipped folder for your operating system *and* the corresponding README. Once you've unzipped the folder on your local computer, you can move it to a location of your choice—or

simply keep it within the downloads folder. Nothing needs to be installed in order for you to begin playing the game.

(If multiple people will be playing TTTB' single-player mode on your computer, I recommend creating a separate copy of the game for each of them. This will make it much easier for each player to keep track of his or her progress.)

## Gameplay instructions

### Starting the game

To launch TTTB, you'll first want to open up your command prompt or terminal, then navigate to the program's build folder (where the game's 'tttb'/'tttb.exe' executable file is located). On my Linux computer, I can accomplish this via the following steps:

1. Press Ctrl + Alt + T to launch my terminal
2. Enter `cd '/home/kjb3/D1V1/Documents/!Dell64docs/Programming/CPP/cpp_tttb/build'` to navigate to the build folder. (This folder path will of course look different on your end.)

On Windows, I would instead press the Windows button, then enter `cmd` to bring up the command prompt. I would then enter `cd` followed by the path to my build folder (perhaps encased in double quotes).

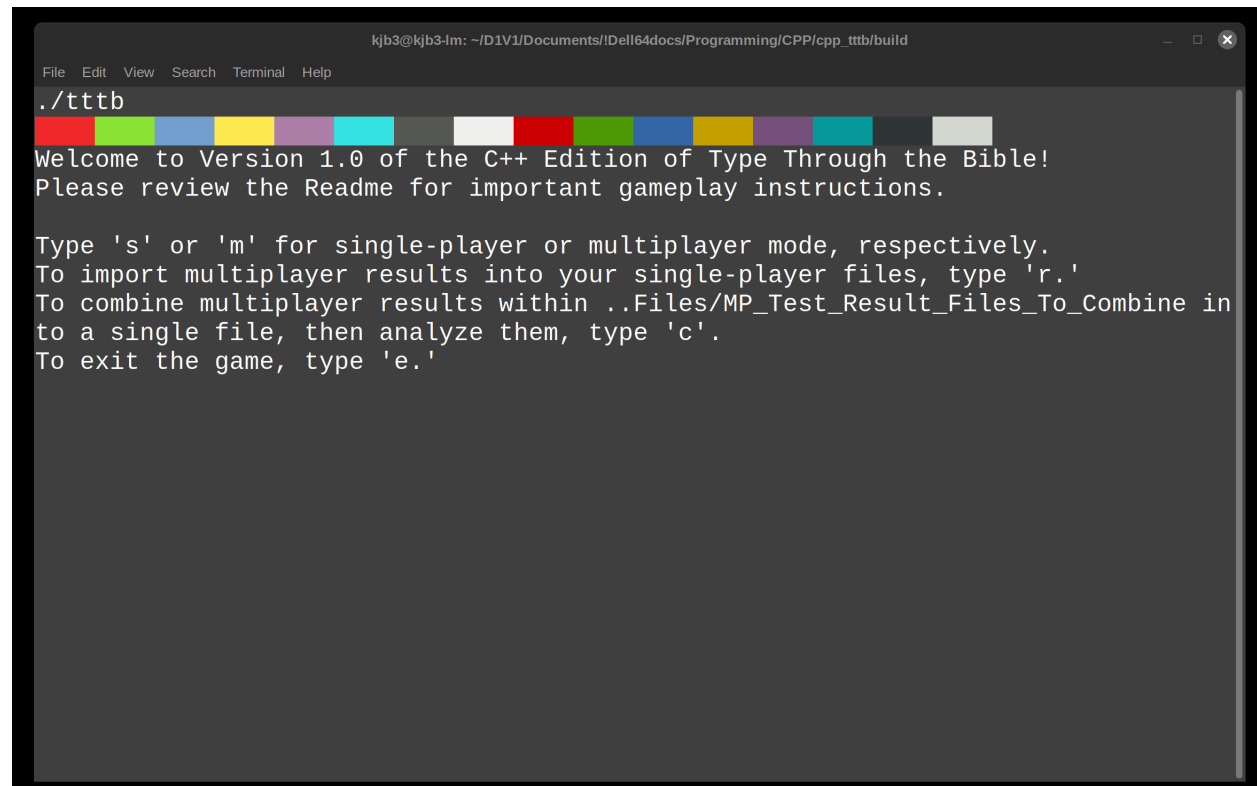
Once you've navigated to this folder, you can launch TTTB by entering the following command (on Linux or OSX):

```
./tttb
```

On Windows, you would instead enter:

```
tttb.exe
```

Once you've completed these steps, you should see the following welcome screen:

A screenshot of a terminal window. The title bar shows the path: `kjb3@kjb3-lm: ~/D1V1/Documents/!Dell64docs/Programming/CPP/cpp_tttb/build`. The terminal has a menu bar with `File Edit View Search Terminal Help`. The prompt is `./tttb`. The output is a colorful ASCII art banner followed by the text: `Welcome to Version 1.0 of the C++ Edition of Type Through the Bible!` and `Please review the Readme for important gameplay instructions.` Below this, instructions are listed: `Type 's' or 'm' for single-player or multiplayer mode, respectively.`, `To import multiplayer results into your single-player files, type 'r.'`, `To combine multiplayer results within ..Files/MP_Test_Result_Files_To_Combine in to a single file, then analyze them, type 'c'.`, and `To exit the game, type 'e.'`

## Playing Type Through the Bible

Type Through the Bible offers both single-player and multiplayer gameplay modes.

**Single-player gameplay** To launch single-player games, press ‘s’ within the welcome screen.

**Multiplayer gameplay** TTTB also lets you combine various copies of results from different computers into a single multiplayer file that you can then analyze.

**Adding multiplayer results to your single-player file** [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. Navigate into TTTB’s `build/` folder.
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 TTTB’s ‘build’ 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!

## Development Notes

I had created an earlier Bible typing game in Python, but given my interest in learning how to apply C++ to create games, I figured that a C++ version of this game would be a great learning opportunity. Although I didn't reference the Python version of this game when writing this code, my experiences with that project certainly influenced the current one.

This project was a lot of fun to work on. While it doesn't feature a GUI, it *was* a good opportunity for me to get reacquainted with C++—and to gain experience with importing and exporting CSV data. I hope that you enjoy playing it as much as I did coding it!

As with my other GitHub projects, I chose not to use generative AI tools when creating Type Through the Bible. I wanted to learn how to perform these tasks in C++ (or in a C++ library) rather than simply learn how to get an AI tool to create them.

This code also makes extensive use of the following open-source libraries:

1. Vincent La's CSV parser (<https://github.com/vincentlaucsb/csv-parser>)
2. CPP-Terminal (<https://github.com/jupyter-xeus/cpp-terminal>)

In addition, this program uses the Catholic Public Domain Version of the Bible that Ronald L. Conte put together. This Bible can be found at <https://sacredbible.org/catholic/> I last updated my local copy of this Bible (whose text does get updated periodically) around June 10, 2025.

## Acknowledgments

I am grateful for Ronald L. Conte Jr. for his work on the [Catholic Public Domain Version](https://sacredbible.org/catholic/) 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!]