

**NAME**

TIMEQUIZ – play a history quiz game

**SYNOPSIS**

timequiz [options]

Called without any option, the game is started with the events that are included in the installed program package. There are not many.

**DESCRIPTION**

The general idea is to put historical events in the right order. In the beginning, a limited number of events (3 at the time of this writing) is presented and you have to indicate the right order of these events. All the following rounds will confront you with just 1 additional event and you must find for the new event the right spot in the previously ordered list, i.e. after one of the listed events or earlier than any of the known events. After each round, the first one inclusive, your performance is evaluated. You can then choose to continue with a new event or, if you like, read a short description of any of the events, already in the list.

The game continues in this way until you push the 'q' button to quit or until all the events that are known to the game, are listed.

**You do not need to know it all**

Beware that to win a round, it is oftentimes not necessary to really know the date of an event. Pure logic and reasonable guesses may be sufficient to find the right spot within a historical context or to exclude events as too far in the past or future of the currently handled event. Reading the descriptive comments to an event may later help you get different events into context.

**OPTIONS**

- a, --add**                      Add a new event
- e, --event [EVENT]**              Name of the new event
- y, --year [YEAR]**              Year of the new event
- b, --background [INFO]**          Background information for the new event
- f, --file [FILE]**              The file, where events are read from or written to
- d, --debug**                  Be verbose
- v, --version**                  Show program version
- h, --help**                  Show a help text

**The Options in detail:**

- a --add**  
This option imposes that **-e**, **-y**, and **-f** be given, too. In consequence, it will be considered superfluous and removed in future versions of the program.
- e --event**  
Name the event to add to the events-file. For this option to take effect, **-a**, **-y** and **-f** are needed, too. Beware to enclose the value with quotes, like in, e.g. **--event "Some guy was born and something broke"**
- y --year**  
Is followed by a year, like **-y 1244**
- b --background**  
Should be used to provide background-information to an event. While this option is ... optional, the information will help solving quizzes and may be educative to the player... do not abuse,

though. Example: `-b "The phenomenon was bearing no significance whatsoever"`

**-f --file**

Name the “events-file” which will be used for either creating the quizzes randomly or adding events to an existing list of events. The file must be readable to serve for a quiz and writable to accept new events. If it does not exist, it is created and some exemplary events are included on top. Example: `--file ~/my_history_quiz`

**-d --debug**

Be verbose. This switches on the output of debug-messages that have been left in the program code. Probably useless.

**-v --version**

Shows the program version

**-h --help**

Shows the option-summary.

## GAME EXAMPLE

Calling timequiz without arguments on the command-line, you are confronted with three arbitrary, historical events, like in the following screen:

Put the following events into the right chronological order (e.g.: 2 3 1).

1. The Roman emperor Diocletian divides the empire in two
2. End of the "War of the mercenaries"
3. Destruction of Nimrod and Ninive by the Medians and Neo-Babylonians

Available: 1, 2, 3)

You type in the three numbers in the order that you consider chronologically correct, like "321" (you do not have to push return after each number). The game will respond by showing you the correct order of events with the corresponding years and an evaluation of your input. You are also prompted to decide if you want to play with a new arbitrary event or first be shown some information about the listed events:

Enter number for more information about an event, 'a' to play with one new, random event, 'q' to quit.

If you push 'a', a new event is shown:

Indicate the event from the previous list, which precedes the following, or '0' (zero) to put it in front:

The Italian painter Lorenzo Lotto dies.

You can now decide, which of the three previous events precedes the death of the Italian painter and either put in the (current) number of that previous event or the number 0 (zero) if you consider that Lorenzo Lotto died well before any of the other events (which would, of course, be false in the example). You have to push the return-button after your input, as it can contain an arbitrary number of ciphers, later in the game.

## OTHER INFORMATION

### Development and source code

timequiz has been written in Ruby. As Ruby is an interpreted programming language, the executable file and all those that it may refer to at one point in time, are themselves the source-files of the current program-version. You can open them in any text-editor to scrutinize the source-code. If you have received the program as a Ruby-gem, you can also decompress a copy of the gem-file

with *tar -x*, then *tar -xzf*.

**Bugs** At one time during the game, the list of already known events is incorrectly wrapped and continues to be.

**License**

timequiz is distributed under the conditions of the GNU General Public License, version 3.

**Author** timequiz has been developed by Michael Uplawski <[michael.uplawski@uplawski.eu](mailto:michael.uplawski@uplawski.eu)>