
Memoria

Release 1.2.0

Jul 03, 2021

Contents

1	Contents	3
1.1	Installation on Linux	3
1.2	Installation on Windows	4
1.3	How to play	4
1.4	Development	10
1.5	Credits	11



Memoria is a simple application to play Memory (Matching) for up to four players.

The program is released under the [GNU General Public License \(GPL\) version 3](#).

1.1 Installation on Linux

1.1.1 Installing Memoria

Before installing Memoria please check the *Dependencies* at the bottom.

After downloading the [source code from GitLab](#) extract the archive to a folder of your choice.

To install Memoria simply run the script `install.sh` in the main directory.

To uninstall the application run the script `uninstall.sh`.

1.1.2 Running uninstalled

If you downloaded the application from GitLab and don't want to install Memoria, simply execute `run.sh` in the main directory.

In this case the game is only available with an English graphical user interface.

1.1.3 Dependencies

Build-time dependencies

As build-time dependencies you need [Meson](#) and [gettext](#) for localization.

Install the dependencies by using the following shell command:

- Arch Linux: `sudo pacman -S meson gettext`
- Debian/Ubuntu: `sudo apt install meson gettext`
- Fedora: `sudo dnf install meson gettext`
- openSUSE: `sudo zypper install meson gettext-tools`

Runtime dependencies

Memoria strongly depends on Python 3 and GTK+ 3:

- GTK+ 3 (≥ 3.16)
- Python 3 (≥ 3.3)
- PyGObject (≥ 3.16)
- Pycairo (≥ 1.10)

Install the runtime dependencies by using the following shell command:

- Arch Linux: `sudo pacman -S python-cairo python-gobject gtk3`
- Debian/Ubuntu: `sudo apt install python3-gi-cairo gir1.2-gtk-3.0`
- Fedora: `sudo dnf install pycairo python3-gobject3 gtk3`
- openSUSE: `sudo zypper install python3-cairo python3-gobject-Gdk libgtk-3-0`

1.2 Installation on Windows

1.2.1 Installing Memoria

To install Memoria just follow the setup wizard of [MemoriaSetup.exe](#).

1.2.2 Building

You can build a Windows executable file and package on your own. Therefor you need to install the development platform [MSYS2](#). Just follow the installation instructions given in the [documentaton of PyGObject](#).

For freezing the Python files into a standalone executable you have to use [PyInstaller](#) (≥ 3.6). To start the build process open a mingw64 terminal and type in the following command in the main directory of Memoria:

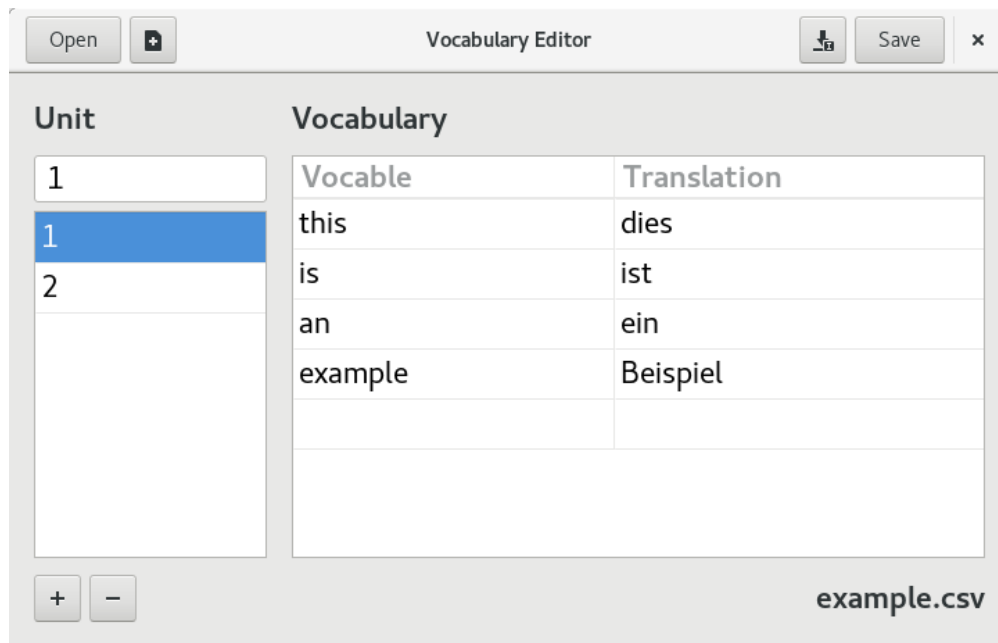
```
pyinstaller setup-win.spec
```

The full package will be created in the folder `dist`.

1.3 How to play

1.3.1 Getting started

To play Memoria you need a vocabulary list. You can use the build-in vocabulary editor, which features easy and intuitive operation, to create new lists or edit existing ones.



Alternatively, you can create and edit vocabulary lists as CSV or TXT files by your own. Please note that the application can only load files with the following data structure:

```
[unit],[vocabulary],[translation]
```

Your vocabulary list should look like this example (separated by comma, semicolon or tab):

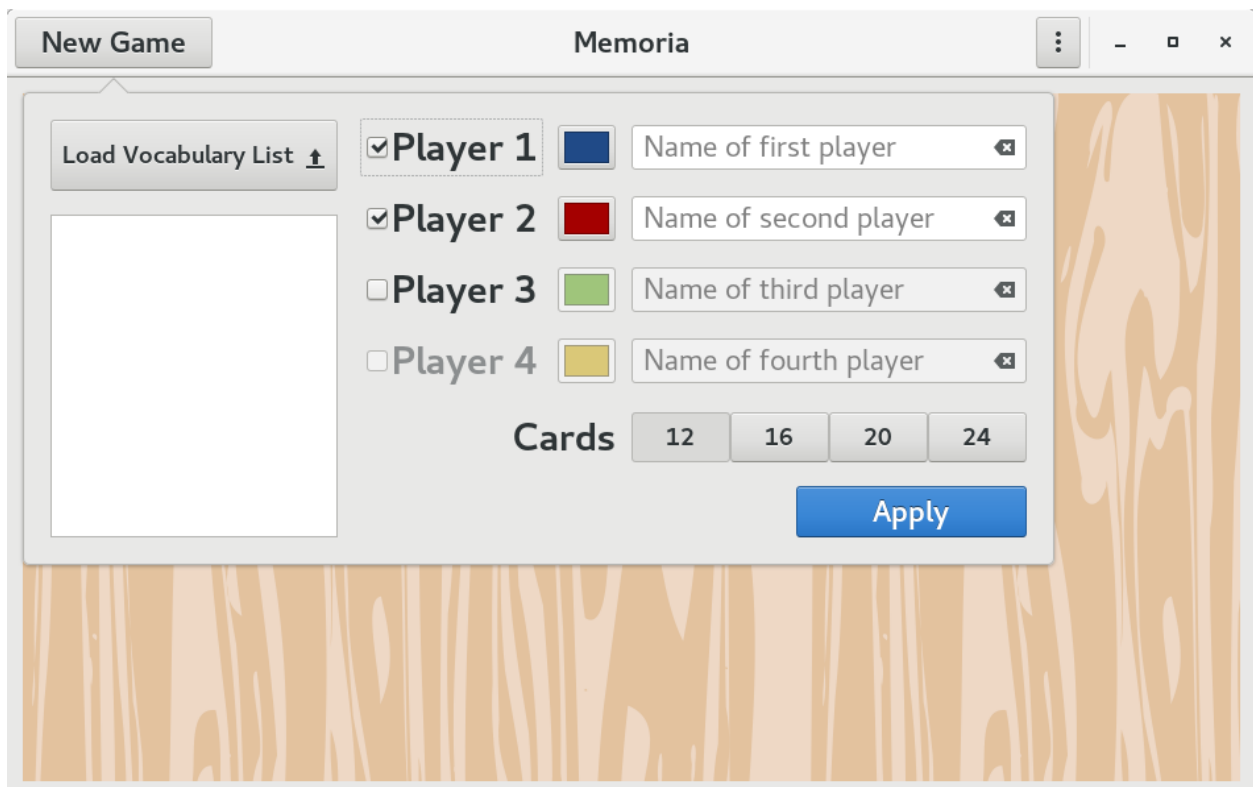
```
1,this,dies
1, is, ist
2,an,ein
2,example,Beispiel
```

1.3.2 Setting up the game

After starting the game you'll see the main window of the application. Open the options window by pressing the button "New Game".

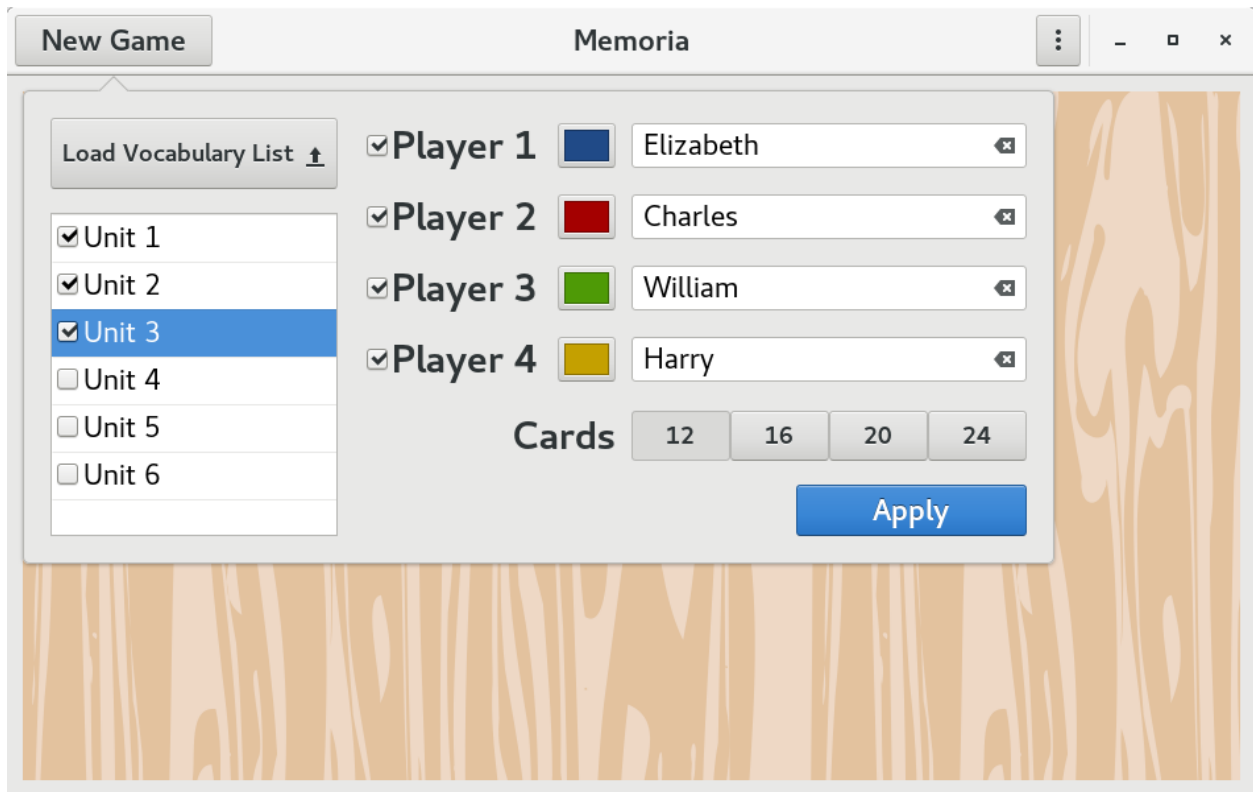


First click on the button “Load Vocabulary List” and navigate to the file you want to open. Now you can choose the units you want your students to repeat.



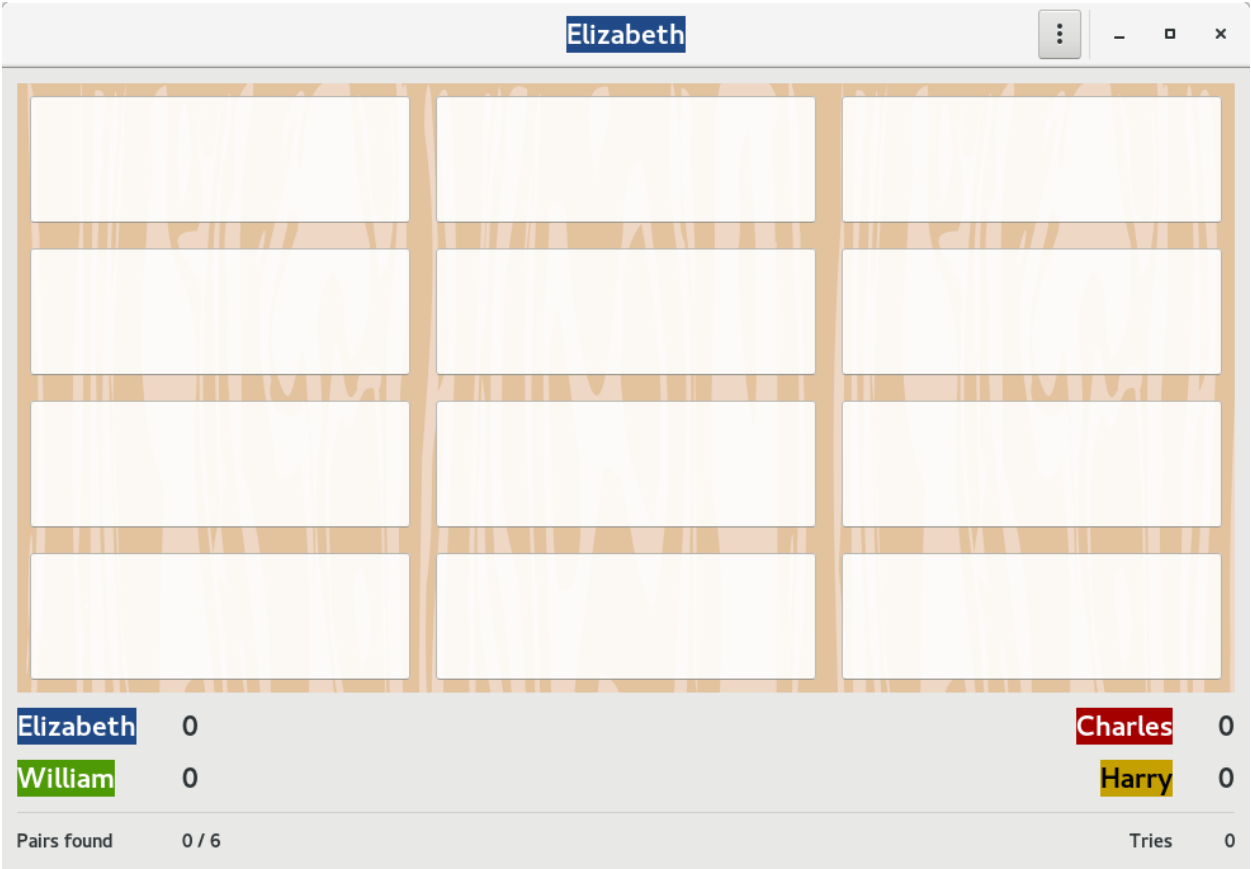
Memoria can be played by up to four students. Select the number of players by clicking on the corresponding check

boxes. Type in the name of each player and change the color if you want. Finally select the number of vocable cards and press the button “Apply”.



1.3.3 Playing the game

You'll find the name of the active player in the header bar, the score and statistics at the bottom of the window.



Try to find a vocable and its translation by clicking on two white rectangels. If you found a matching pair, the rectangels are highlighted by the player’s color and you can go on. Otherwise it’s the turn of the next player.

Next player (1)

Haus

Baum

house

flower

Elizabeth

0

William

0

Pairs found

1 / 6

Charles

0

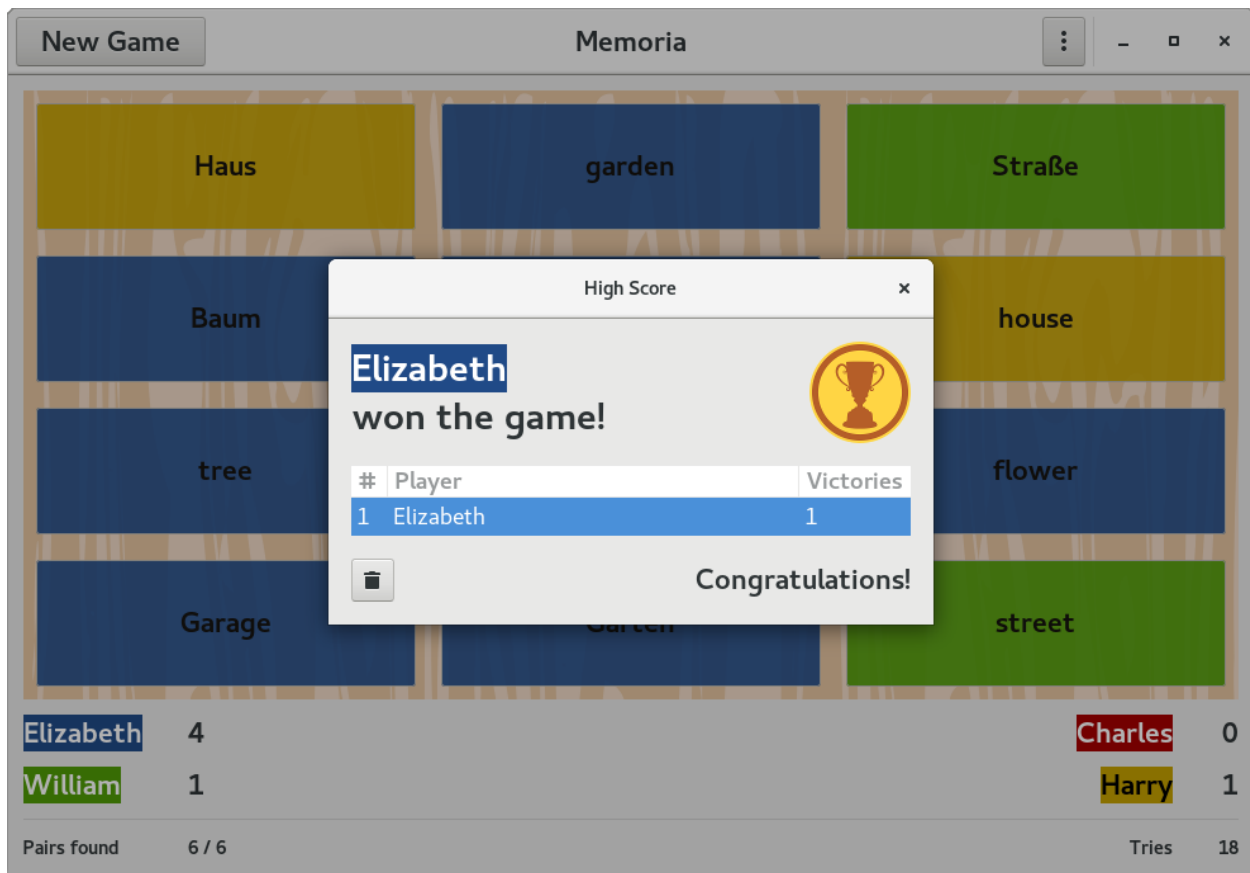
Harry

1

Tries

9

The player who found most matching pairs wins the game.



1.4 Development

1.4.1 Roadmap

The current version of Memoria is 1.2.0.

For future releases the following features are planned:

- better support for HiDPI
- port to GTK 4

1.4.2 Bugs

Known Bugs

At the moment no bugs are known.

Filing A Bug

If you've found a bug in Memoria, please head over to GitLab and [file a report](#). Filing bugs helps improve the software for everyone.

1.4.3 Source Code

The full source code of Memoria is hosted on [GitLab](#).

1.5 Credits

Memoria is developed by Thomas Dähnrich.

Special thanks go to:

- Joe Hamilton (for his fabulous [simpleaudio](#) package)
- the developers of [MSYS2](#), [PyInstaller](#) and [Inno Setup](#)
- the Latin students of the [Vicco-von-Bülow-Gymnasium Falkensee](#) (for being enthusiastic beta testers)