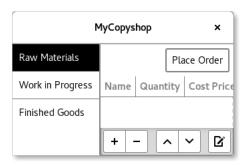
Handout 5: Meson 1



We want to use Meson so that we can easily build, test and distribute bigger projects.

This means our project should have a clear folder structure. We create for example as a foundation

- in a newly created empty project directory
 - a text file called README which is meant to contain basic explanations what the software does, how to install it and what its license is.
 - a file AUTHORS which is meant to contain the names of all developers later on,
 - a NEWS file which is meant to briefly list changes between software versions that are important to users,
 - a file called COPYING or LICENSE containing the software license (software under multiple licenses stores them in one file each in a LICENSES directory instead, cf. https://reuse.software),
 - a currently still empty text file called meson.build,
 - a subdirectory called src or mycopyshop or similar
 - * with a file main.c inside which will later contain the main() function
 - * as well as another meson.build file.

Text files written in *Markdown* may additionally use the .md file extension. The src directory will contain all the program's C source files later on.

This should be added to the main project folder's meson.build file:

```
project('mycopyshop', 'c') gtkdep = dependency('gtk+-3.0', version : '>=_{\square}3.22') subdir('src')
```

Handout 5: Meson 2

In real projects project() should also state the program's license and version, see http://mesonbuild.com. The GTK+ version needed has been specified as 3.22 in this example.

src/meson.build should contain the following:

```
executable ('copyshop', 'main.c',
  dependencies : gtkdep,
  install : true,
  gui_app : true)
```

Done! We now want to store our progress in a Git repository to make sure we won't lose it. To do so we open a terminal in the main project directory and run these commands:

```
$ git init
$ git add README AUTHORS NEWS COPYING meson.build
  src/main.c src/meson.build
```

Using git we can easily manage which files are part of the project. All foreign files can be quickly removed using git clean -df. git status shows the status of the files in the project directory. Most people use Git for saving and managing changes to the project in one version ("commit") each. This way - among other advantages - multiple developers can cooperate efficiently.

ily lost, we build and install the pro- for all users as well: ject for a single user:

Now that changes are no longer eas- Administrators can install the project

```
$ mkdir builddir
$ mkdir builddir
                                  $ cd builddir
$ cd builddir
                                  $ meson ..
$ meson .. --prefix=$HOME/.local $ ninja
$ ninja install
                                  $ sudo ninja install
```

If the project requires Glade files or similar to be loaded, we need to either include them as a resource within the program or supply their file path to the C code in a configuration file. For the latter we need a config.h file; for an example on how to create one, see the next handout or the example here:

https://pelzflorian.de/git/gui-prog-gtk/tree/bin/copyshop/meson.build