Tui Widgets

Ein Baukasten für Terminal-Anwendungen

PRESENTED BY:

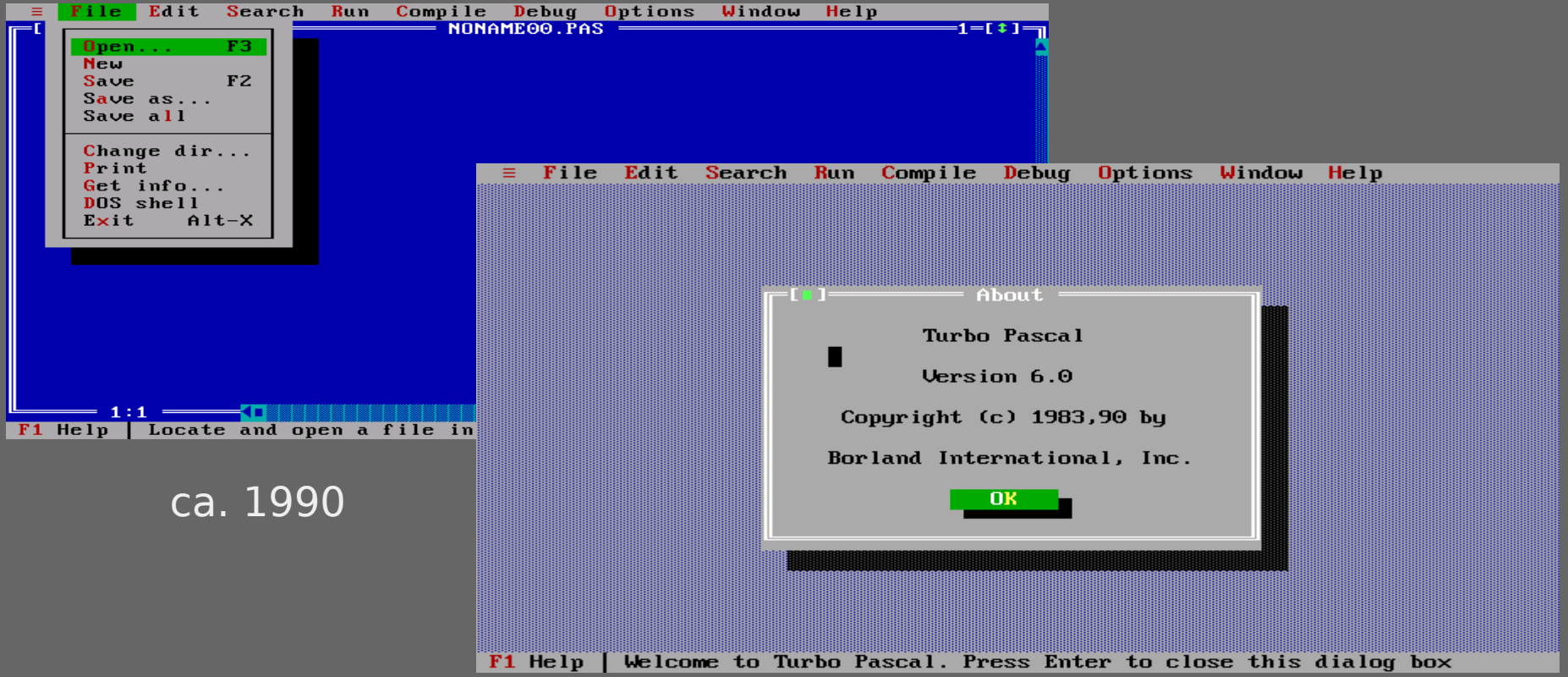
Christoph Hüffelmann

Martin Hostettler

Warum?

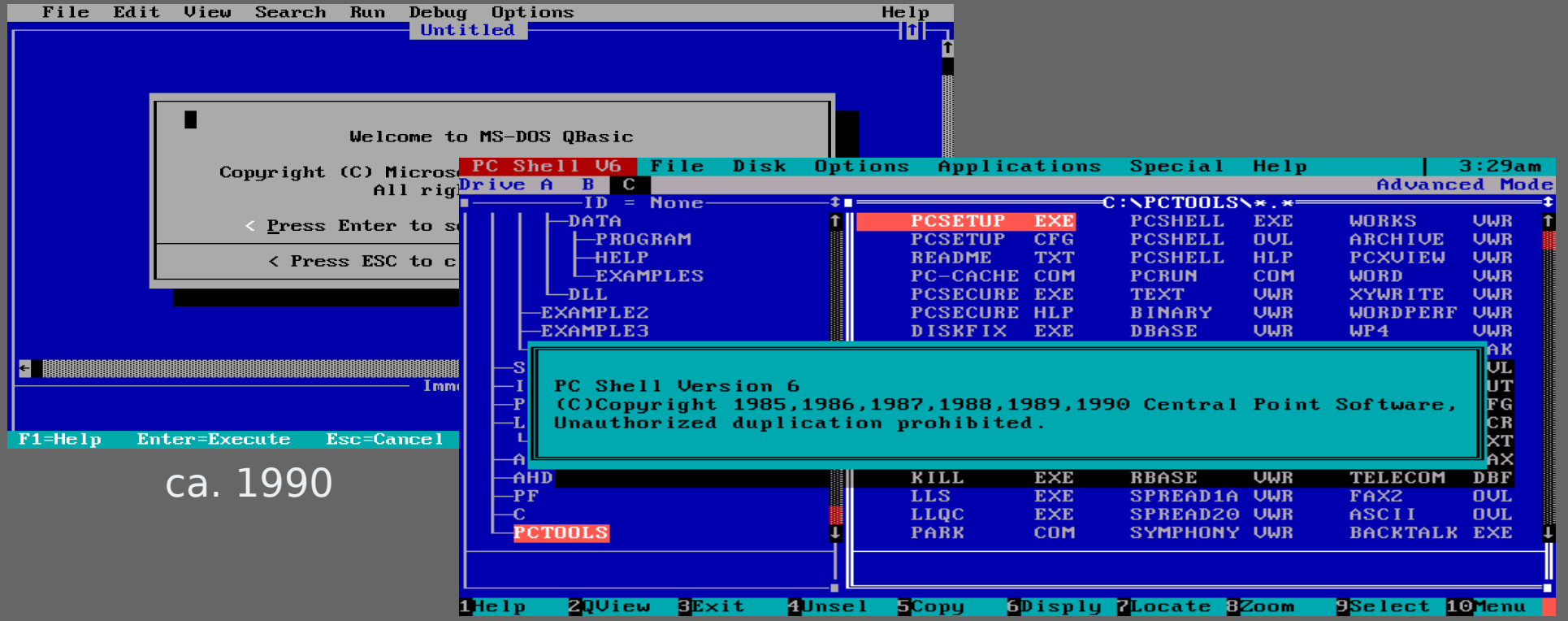
- TUI mit
  - Reusable Components
  - Event Loop
  - Unicode support
  - “normale” Keyboard shortcuts / Fenster
- Fokus auf moderne Terminals
  - grob was UTF-8 kann
- Robustes Keyboard Handling

Inspirationen



ca. 1990

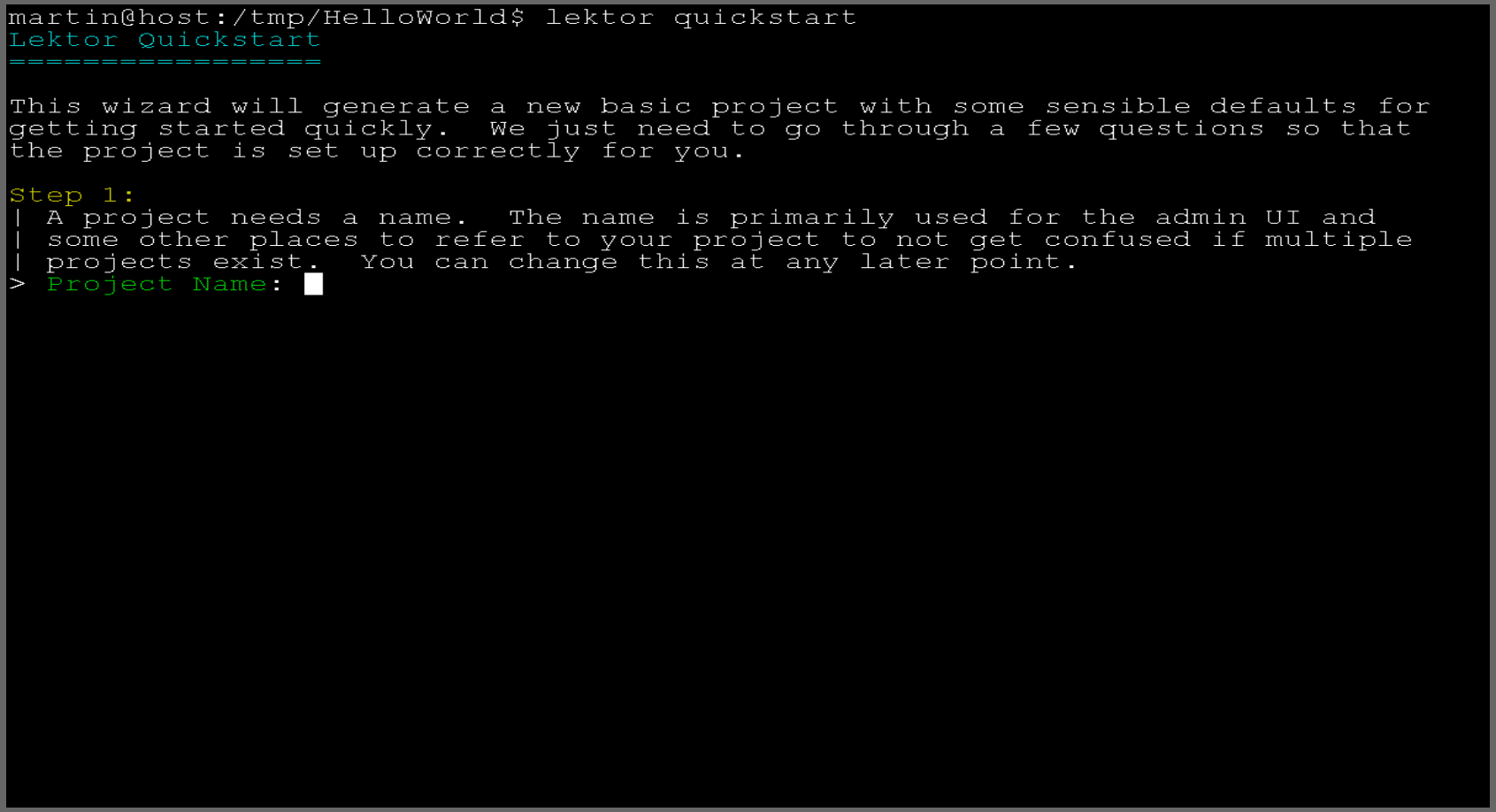
Inspirationen



ca. 1990

CLI / TUI / GUI

CLI / TUI / GUI


martin@host:/tmp/HelloWorld$ lektor quickstart
Lektor Quickstart
==========
This wizard will generate a new basic project with some sensible defaults for
getting started quickly. We just need to go through a few questions so that
the project is set up correctly for you.
Step 1:
| A project needs a name. The name is primarily used for the admin UI and
| some other places to refer to your project to not get confused if multiple
| projects exist. You can change this at any later point.
> Project Name:

CLI / TUI / GUI

```
martin@host:/tmp/HelloWorld$ lektor quickstart
Lektor Quickstart
=================

This wizard will generate a new basic project with some sensible defaults for
getting started quickly.  We just need to go through a few questions so that
the project is set up correctly for you.

Step 1:
| A project needs a name.  The name is primarily used for the admin UI and
| some other places to refer to your project to not get confused if multiple
| projects exist.  You can change this at any later point.
> Project Name: Example

Step 2:
| This is the path where the project will be located.  You can move a
| project around later if you do not like the path.  If you provide a
| relative path it will be relative to the working directory.
> Project Path [/tmp/HelloWorld/Example]:
```

CLI / TUI / GUI

```
martin@host:/tmp/HelloWorld$ lektor quickstart
Lektor Quickstart
=================

This wizard will generate a new basic project with some sensible defaults for
getting started quickly.  We just need to go through a few questions so that
the project is set up correctly for you.

Step 1:
| A project needs a name.  The name is primarily used for the admin UI and
| some other places to refer to your project to not get confused if multiple
| projects exist.  You can change this at any later point.
> Project Name: Example

Step 2:
| This is the path where the project will be located.  You can move a
| project around later if you do not like the path.  If you provide a
| relative path it will be relative to the working directory.
> Project Path [/tmp/HelloWorld/Example]:

Step 3:
| Do you want to generate a basic blog module?  If you enable this the
| models for a very basic blog will be generated.
> Add Basic Blog [Y/n]:
```

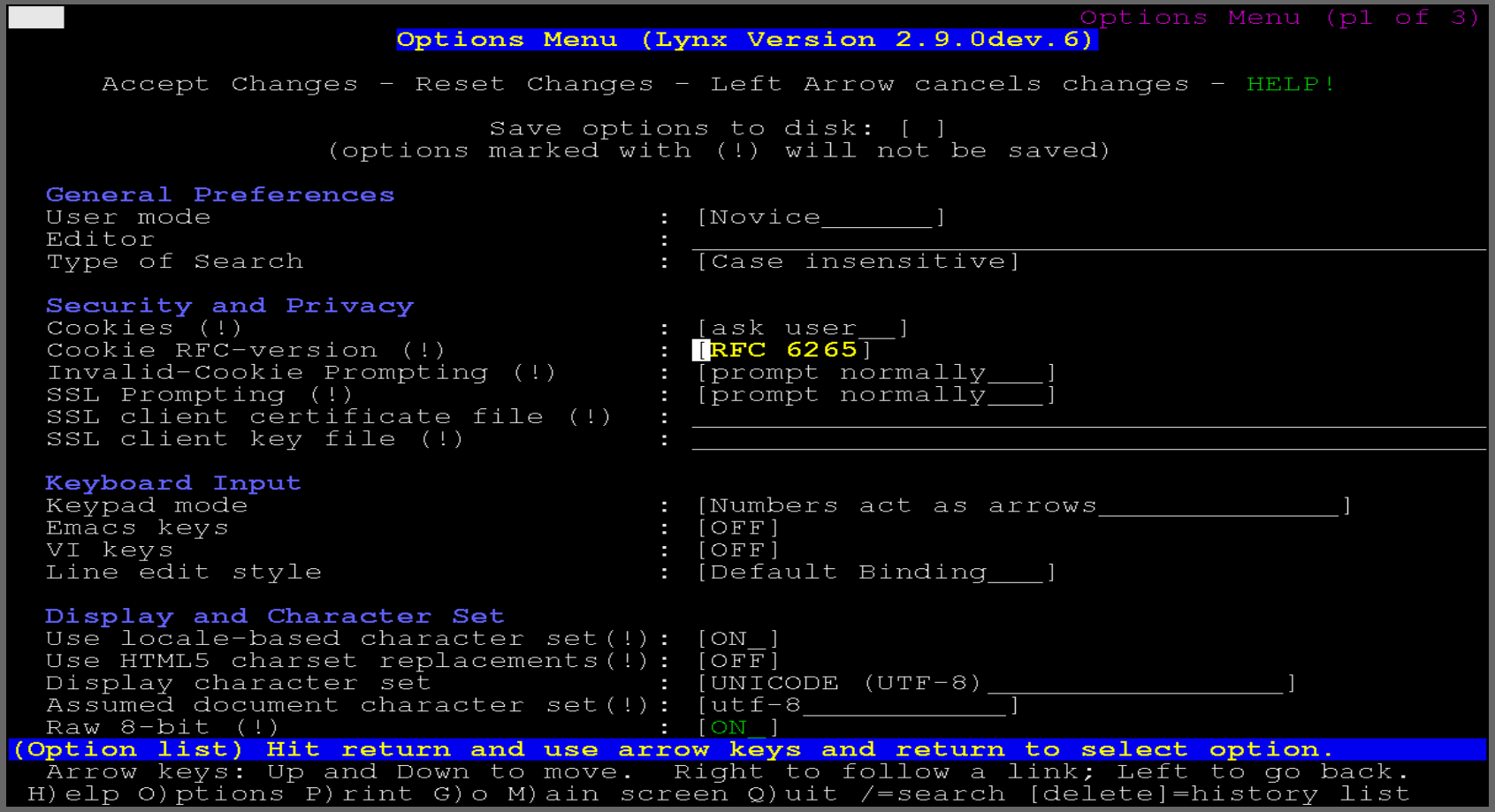
CLI / TUI / GUI

```
martin@host:/tmp/HelloWorld$ lektor quickstart
Lektor Quickstart
=================

This wizard will generate a new basic project with some sensible defaults for
getting started quickly.  We just need to go through a few questions so that
the project is set up correctly for you.

Step 1:
| A project needs a name.  The name is primarily used for the admin UI and
| some other places to refer to your project to not get confused if multiple
| projects exist.  You can change this at any later point.
> Project Name: Example

Step 2:
| This is the path where the project will be located.  You can move a
| project around later if you do not like the path.  If you provide a
| relative path it will be relative to the working directory.
> Project Path [/tmp/HelloWorld/Example]:

Step 3:
| Do you want to generate a basic blog module?  If you enable this the
| models for a very basic blog will be generated.
> Add Basic Blog [Y/n]: n

Step 4:
| Your name.  This is used in a few places in the default template to refer
| to in the default copyright messages.
> Author Name [,,,]:
```

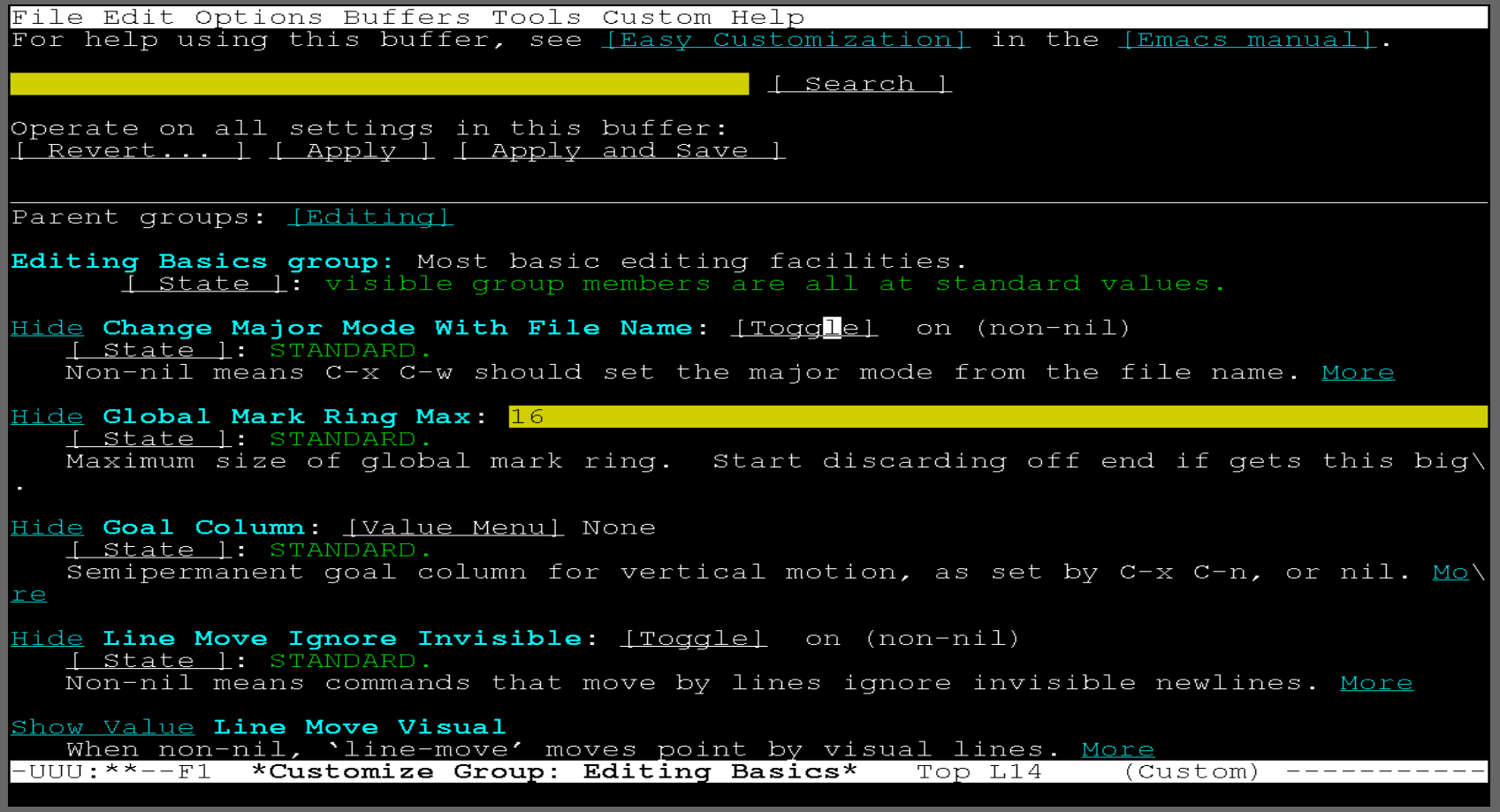
CLI / TUI / GUI

```
martin@host:/tmp/HelloWorld$ lektor quickstart
Lektor Quickstart
=================

This wizard will generate a new basic project with some sensible defaults for
getting started quickly.  We just need to go through a few questions so that
the project is set up correctly for you.

Step 1:
| A project needs a name.  The name is primarily used for the admin UI and
| some other places to refer to your project to not get confused if multiple
| projects exist.  You can change this at any later point.
> Project Name: Example

Step 2:
| This is the path where the project will be located.  You can move a
| project around later if you do not like the path.  If you provide a
| relative path it will be relative to the working directory.
> Project Path [/tmp/HelloWorld/Example]:

Step 3:
| Do you want to generate a basic blog module?  If you enable this the
| models for a very basic blog will be generated.
> Add Basic Blog [Y/n]: n

Step 4:
| Your name.  This is used in a few places in the default template to refer
| to in the default copyright messages.
> Author Name [,,,]: Someone

That's all. Create project? [Y/n] 
```

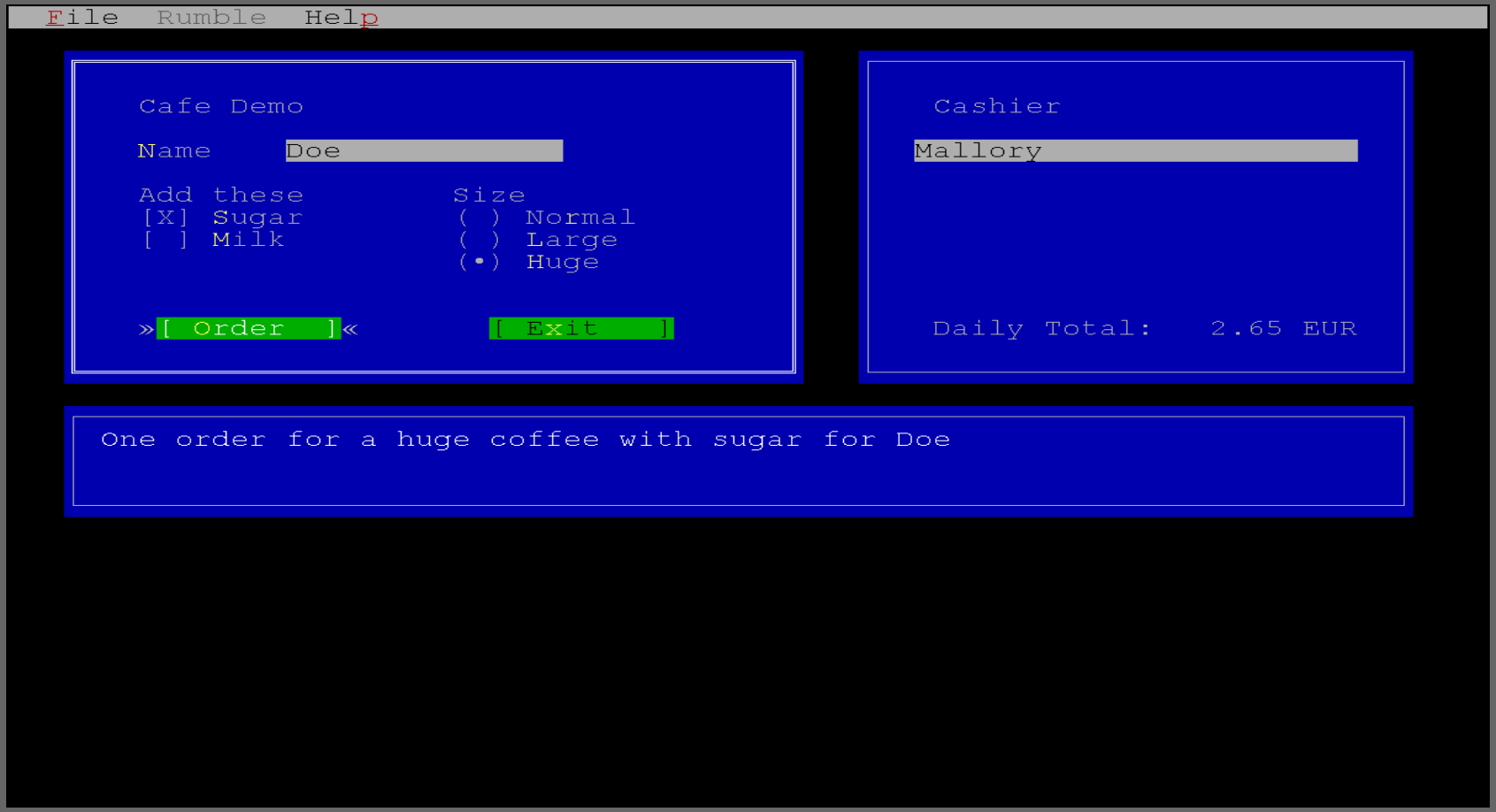
CLI / TUI / GUI


Options Menu (p1 of 3)
Options Menu (Lynx Version 2.9.0dev.6)
Accept Changes - Reset Changes - Left Arrow cancels changes - HELP!
Save options to disk: [ ]
(options marked with (!) will not be saved)
General Preferences
User mode : [Novice_____]
Editor :
Type of Search : [Case insensitive]
Security and Privacy
Cookies (!) : [ask user__]
Cookie RFC-version (!) : [RFC 6265]
Invalid-Cookie Prompting (!) : [prompt normally___]
SSL Prompting (!) : [prompt normally___]
SSL client certificate file (!) :
SSL client key file (!) :
Keyboard Input
Keypad mode : [Numbers act as arrows_____]
Emacs keys : [OFF]
VI keys : [OFF]
Line edit style : [Default Binding___]
Display and Character Set
Use locale-based character set(!): [ON_]
Use HTML5 charset replacements(!): [OFF]
Display character set : [UNICODE (UTF-8)_____]
Assumed document character set(!): [utf-8_____]
Raw 8-bit (!) : [ON]
(Option list) Hit return and use arrow keys and return to select option.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

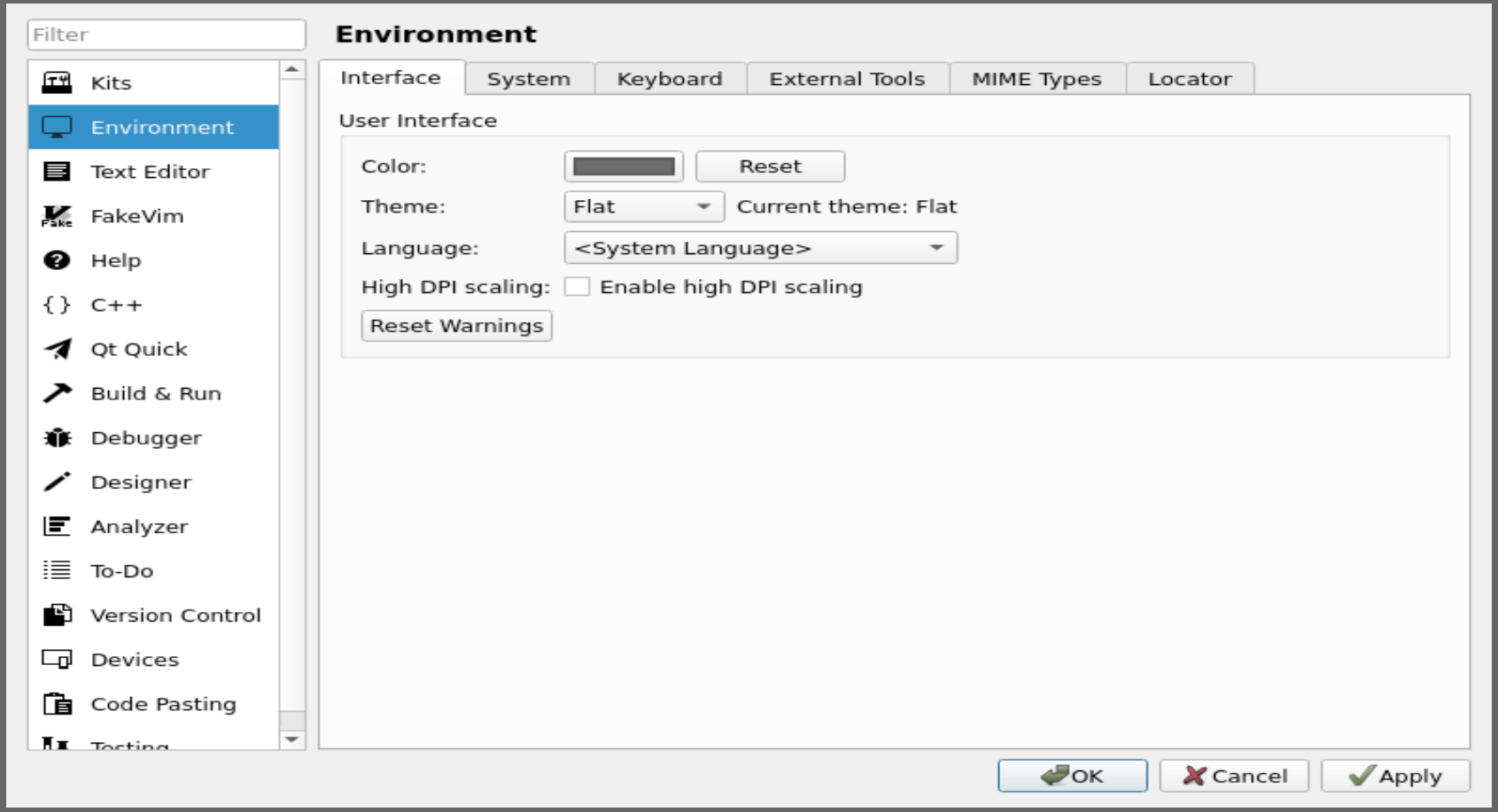
CLI / TUI / GUI

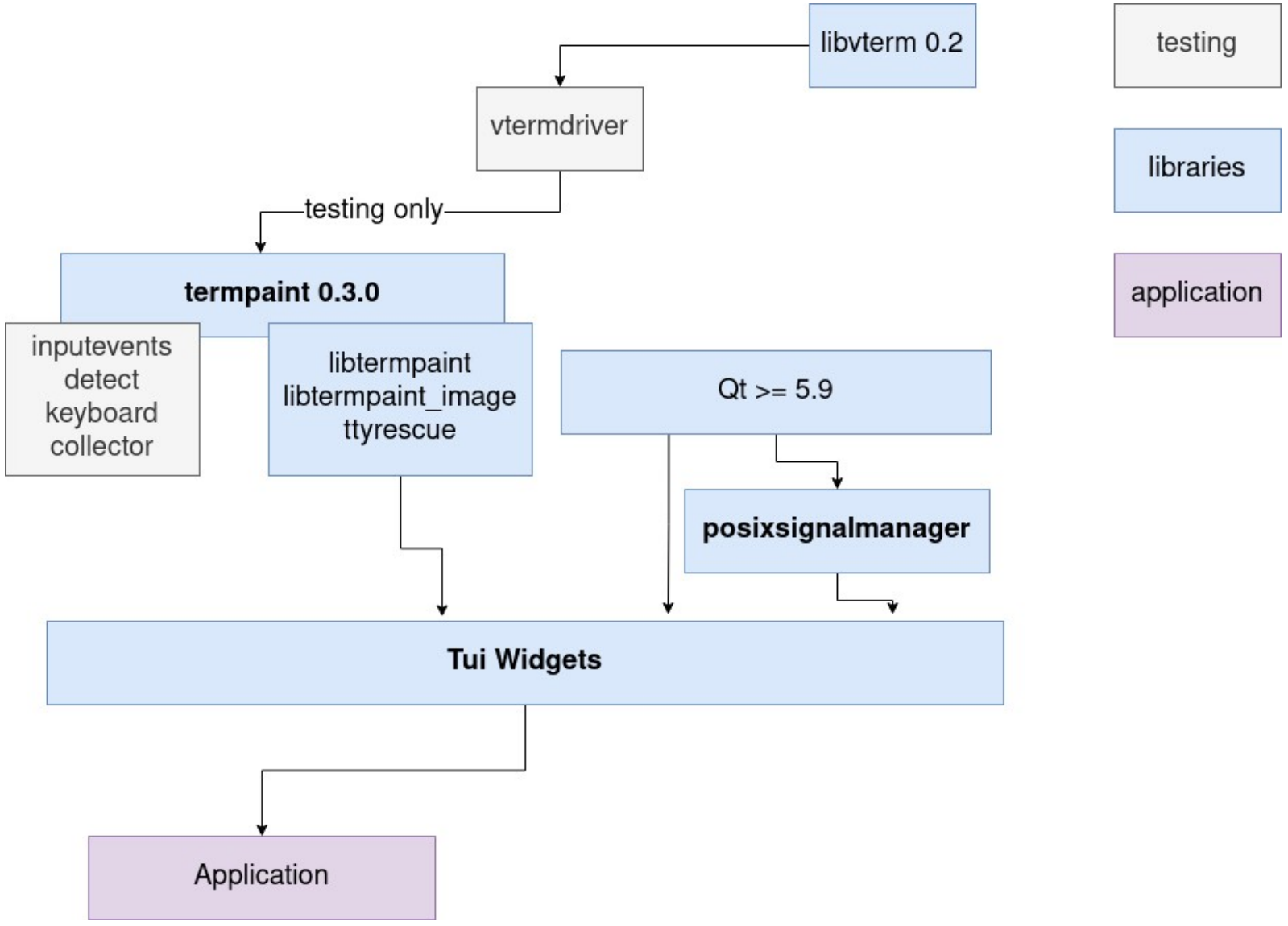

File Edit Options Buffers Tools Custom Help
For help using this buffer, see [Easy Customization] in the [Emacs manual].
[Search]
Operate on all settings in this buffer:
[Revert...] [Apply] [Apply and Save]
Parent groups: [Editing]
Editing Basics group: Most basic editing facilities.
[State]: visible group members are all at standard values.
Hide Change Major Mode With File Name: [Toggle] on (non-nil)
[State]: STANDARD.
Non-nil means C-x C-w should set the major mode from the file name. More
Hide Global Mark Ring Max: 16
[State]: STANDARD.
Maximum size of global mark ring. Start discarding off end if gets this big\
.
Hide Goal Column: [Value Menu] None
[State]: STANDARD.
Semipermanent goal column for vertical motion, as set by C-x C-n, or nil. Mo\
re
Hide Line Move Ignore Invisible: [Toggle] on (non-nil)
[State]: STANDARD.
Non-nil means commands that move by lines ignore invisible newlines. More
Show Value Line Move Visual
When non-nil, `line-move' moves point by visual lines. More
-UUU:**--F1 *Customize Group: Editing Basics* Top L14 (Custom) ----------

CLI / TUI / GUI

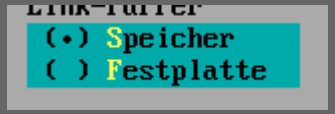

File Rumble Help
Cafe Demo
Name Doe
Add these
[X] Sugar
[ ] Milk
Size
( ) Normal
( ) Large
(•) Huge
»[Order]«
[Exit]
Cashier
Mallory
Daily Total: 2.65 EUR
One order for a huge coffee with sugar for Doe

CLI / TUI / GUI


Filter
Kits
Environment
Text Editor
FakeVim
Help
C++
Qt Quick
Build & Run
Debugger
Designer
Analyzer
To-Do
Version Control
Devices
Code Pasting
Environment
Interface
System
Keyboard
External Tools
MIME Types
Locator
User Interface
Color:
Reset
Theme:
Flat
Current theme: Flat
Language:
<System Language>
High DPI scaling:
Enable high DPI scaling
Reset Warnings
OK
Cancel
Apply


libvterm 0.2
vtermdriver
testing only
termpaint 0.3.0
inputevents
detect
keyboard
collector
libtermpaint
libtermpaint_image
ttyrescue
Qt >= 5.9
posixsignalmanager
Tui Widgets
Application
testing
libraries
application

Tui Widgets

- Basierend auf Qt5 (≥5.9)*
  - Betriebsystem Abstraktionen und Event-Loop
  - Objectmodell
- Termpaint für Terminalansteuerung
- Fertige Komponenten im 

Stil
- C++ Library, namespace Tui::

* Für Qt6 Support fehlt eine stabile API für restacking von Children in Qt

Prinzipien

- Look und Feel mehr wie GUI Anwendungen
- Farben abstrahiert durch Palette
  - für Entwickler einstellbar
- Auch terminals mit weniger Features supporten
  - z.B. monochrome, kleiner Zeichensatz

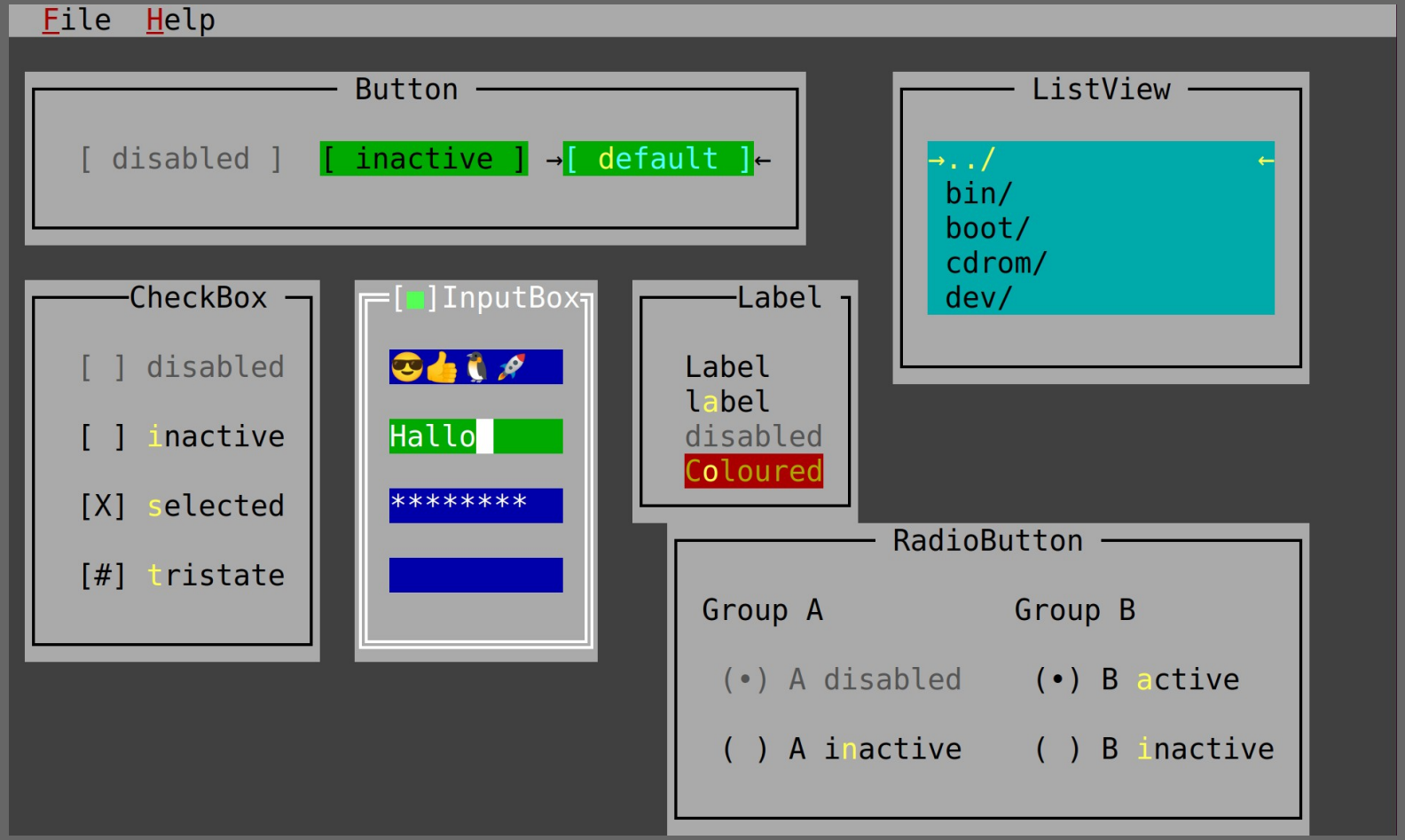
Prinzipien

- ABI Stabile Library (wip)
- Möglichst modular, rein interne Schnittstellen vermeiden
- Global state vermeiden
  - z.B. support für mehere Terminals gleichzeitig

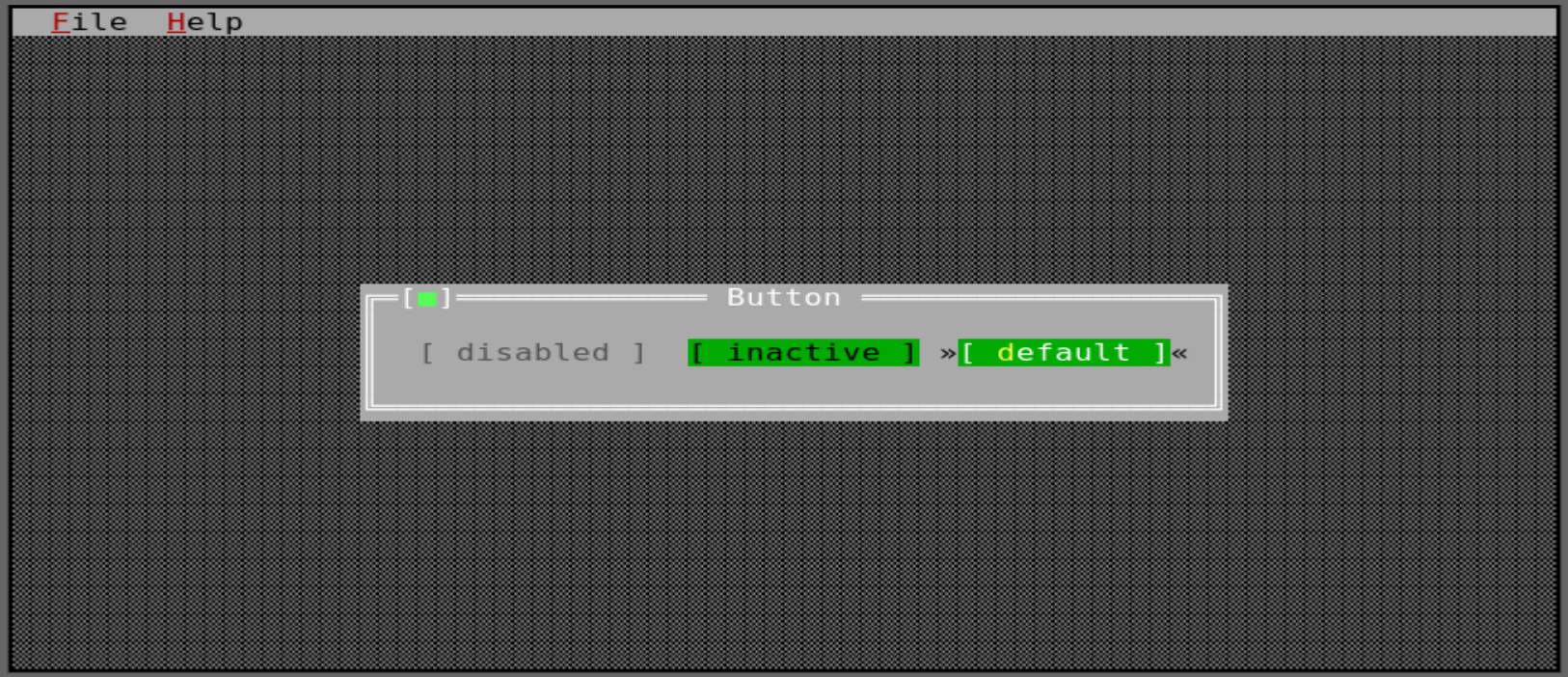
Widgets

- (demo)

Widgets


File Help
Button
[disabled] [inactive] →[default]←
ListView
→../ ←
bin/
boot/
cdrom/
dev/
CheckBox
[ ] disabled
[ ] inactive
[X] selected
[#] tristate
[■]InputBox
Hallo
****
Label
Label
label
disabled
Coloured
RadioButton
Group A
(•) A disabled
( ) A inactive
Group B
(•) B active
( ) B inactive

Button's

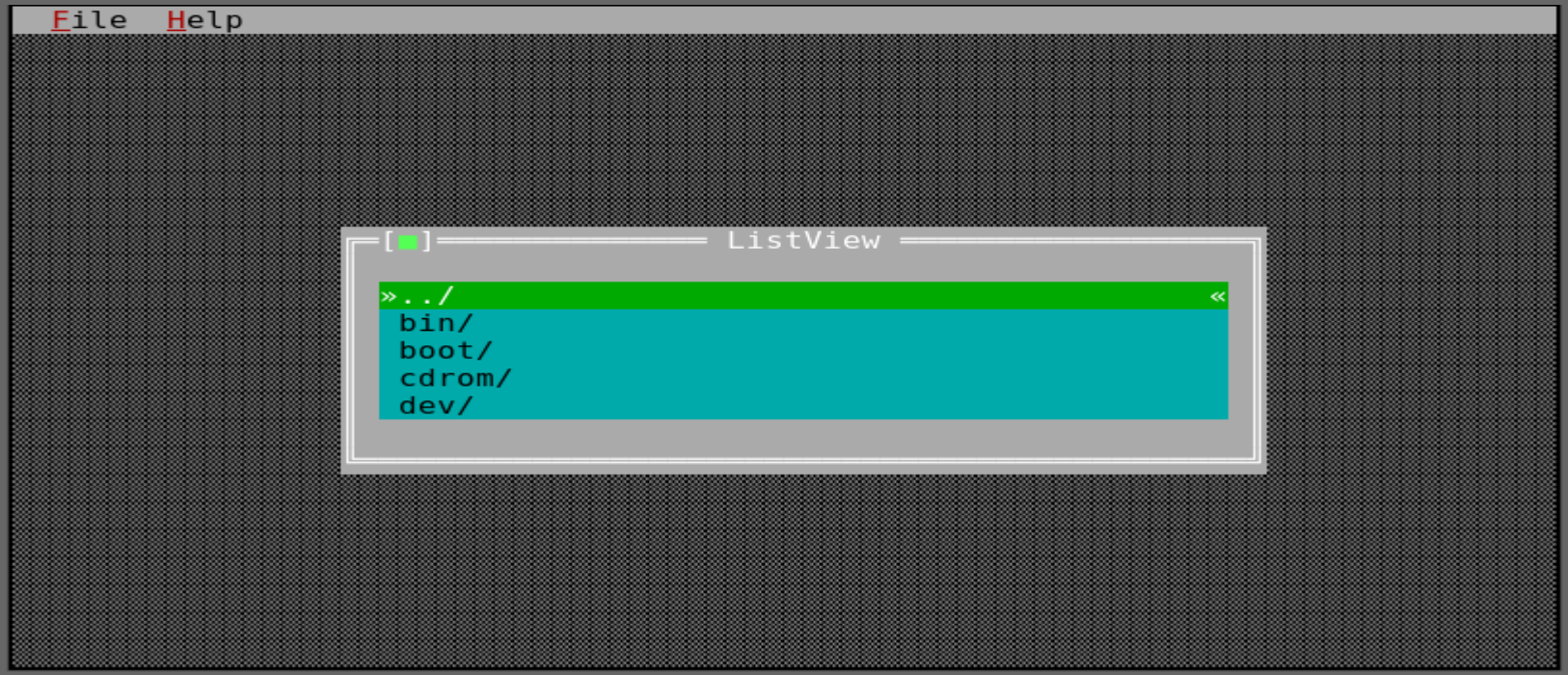


CheckBox

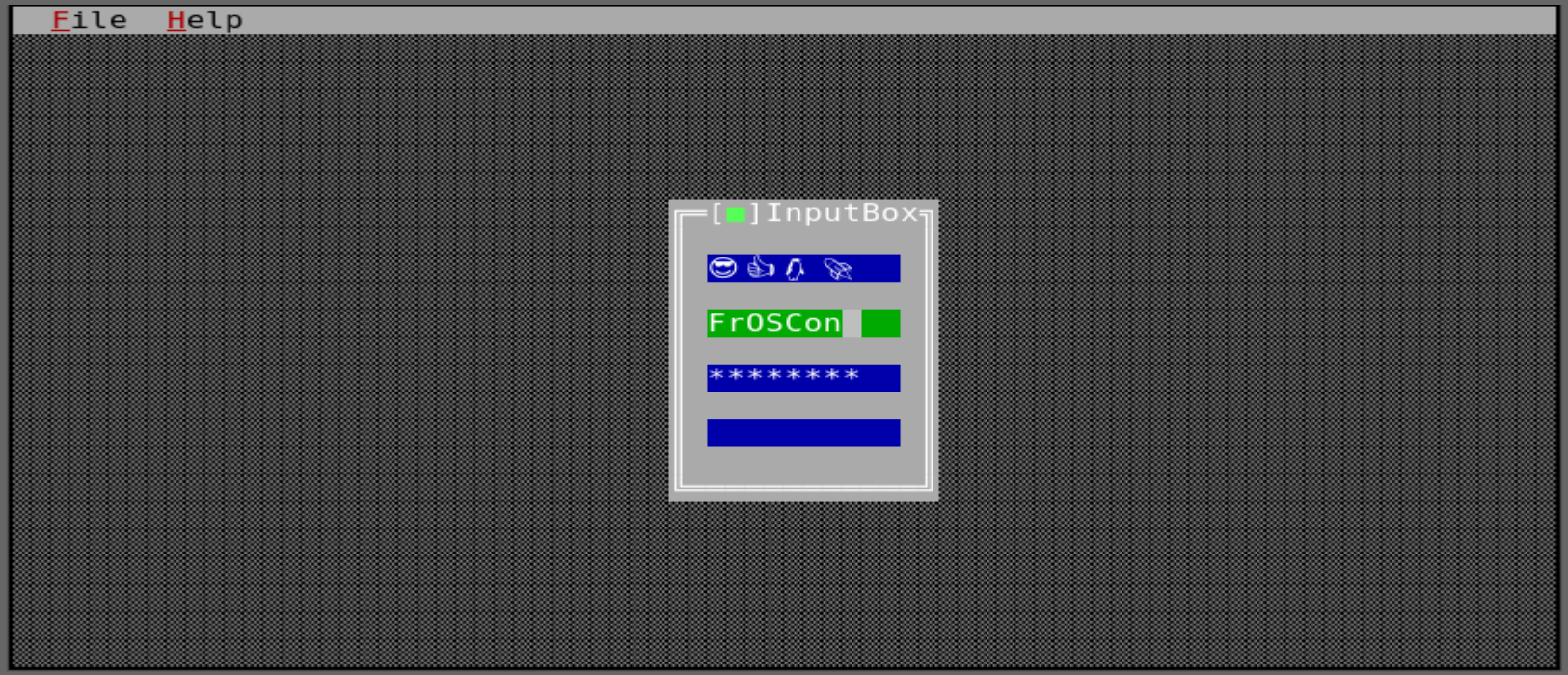
RadioButton

Label

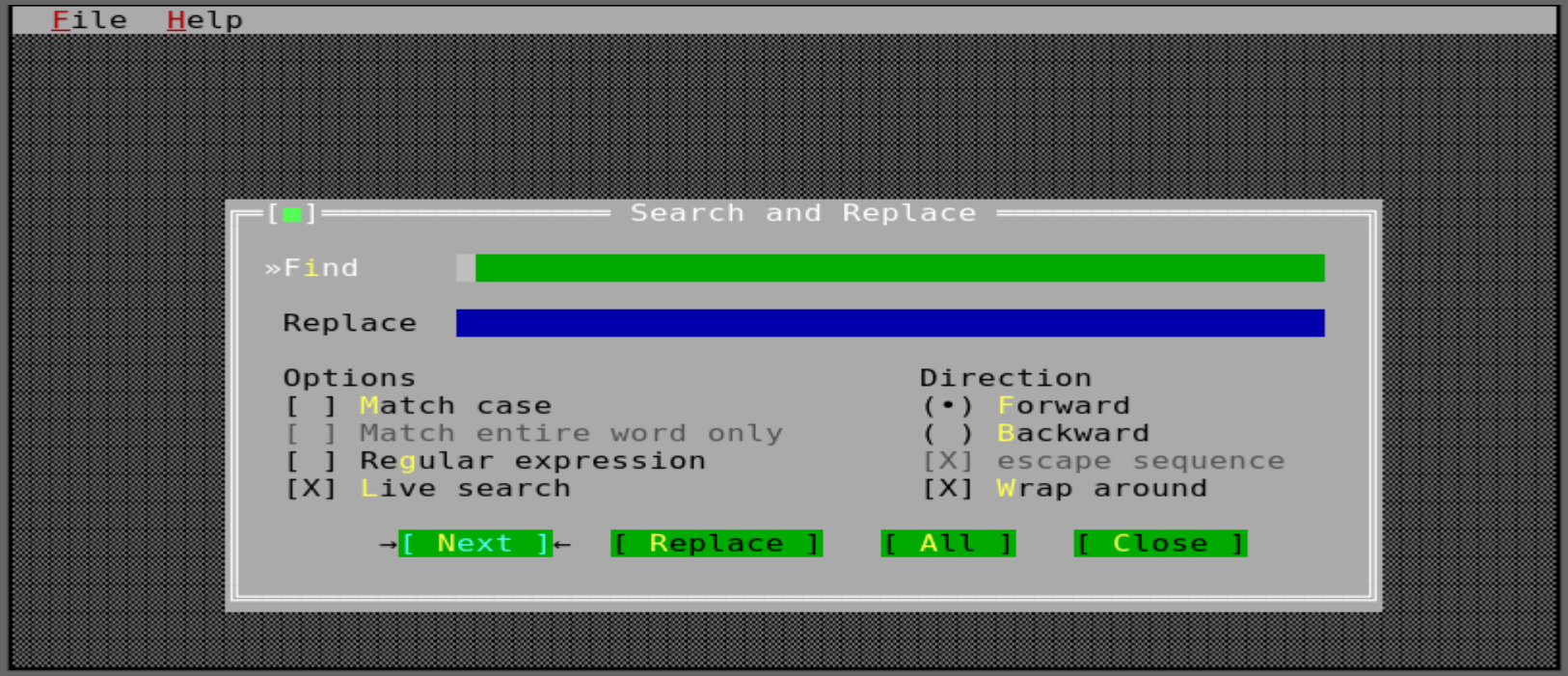
ListView



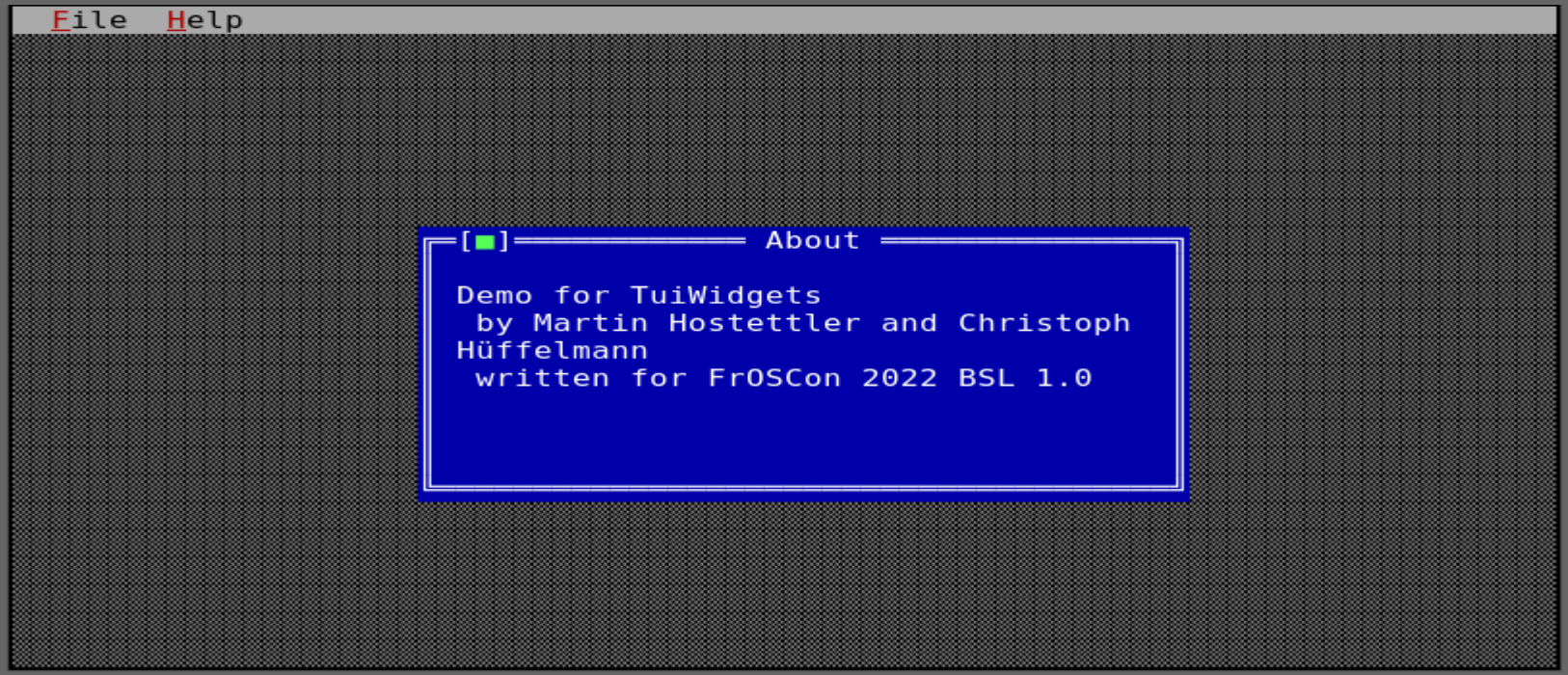
InputBox



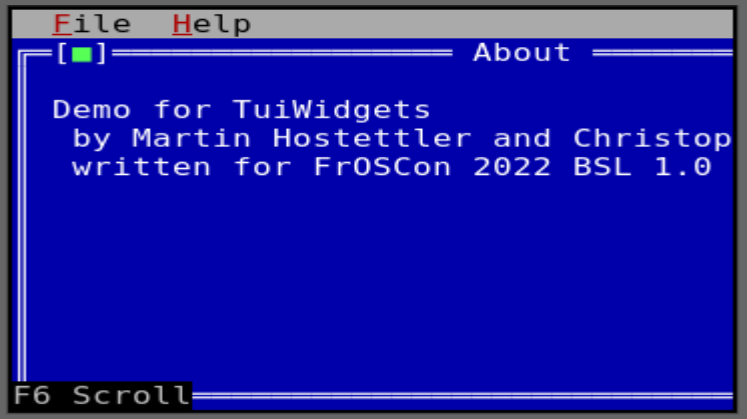
H/VBoxen


File Help
Search and Replace
»Find
Replace
Options
[ ] Match case
[ ] Match entire word only
[ ] Regular expression
[X] Live search
Direction
(•) Forward
( ) Backward
[X] escape sequence
[X] Wrap around
→[Next]←
[Replace]
[All]
[Close]

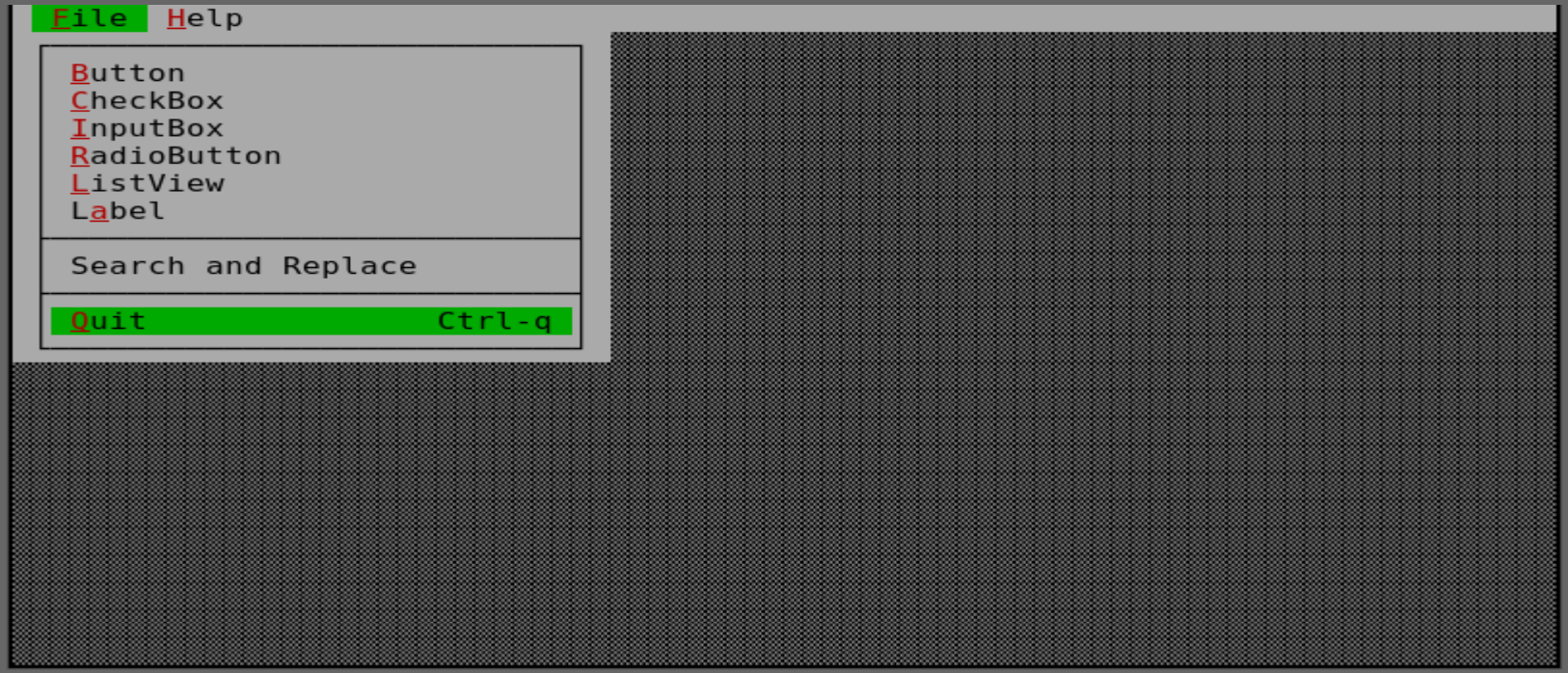
Window


File Help
About
Demo for TuiWidgets
by Martin Hostettler and Christoph Hüffelmann
written for FrOSCon 2022 BSL 1.0

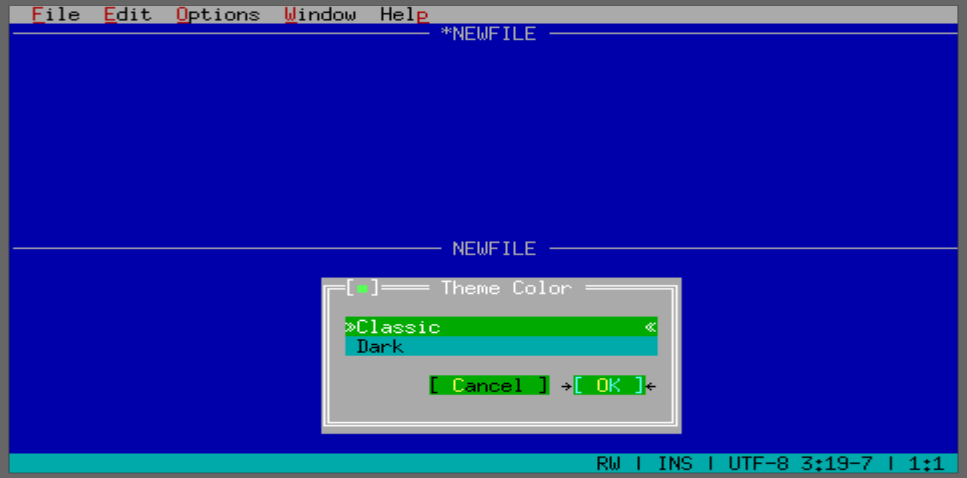
F6 Modus


File Help
[■] About
Demo for TuiWidgets
by Martin Hostettler and Christop
written for FrOSCon 2022 BSL 1.0
F6 Scroll

MenueBar und Menue

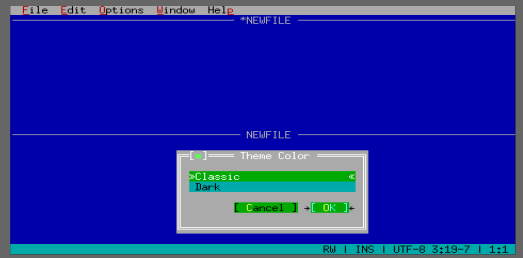


Layering


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel] →[OK]←
RW | INS | UTF-8 3:19-7 | 1:1

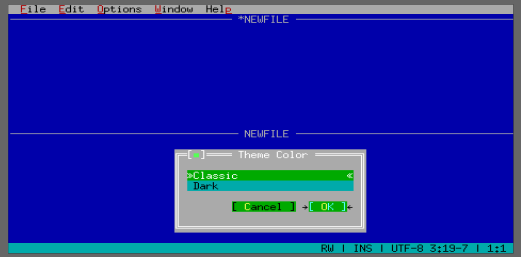

root
menu
window 1
window 2
dialog
listview
cancel
ok
statusbar

Layering


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel] [OK]
RW | INS | UTF-8 3:19-7 | 1:1

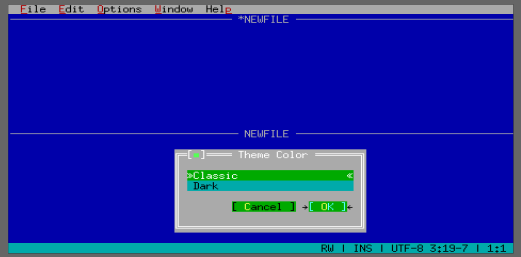
Layering


File Edit Options Window Help

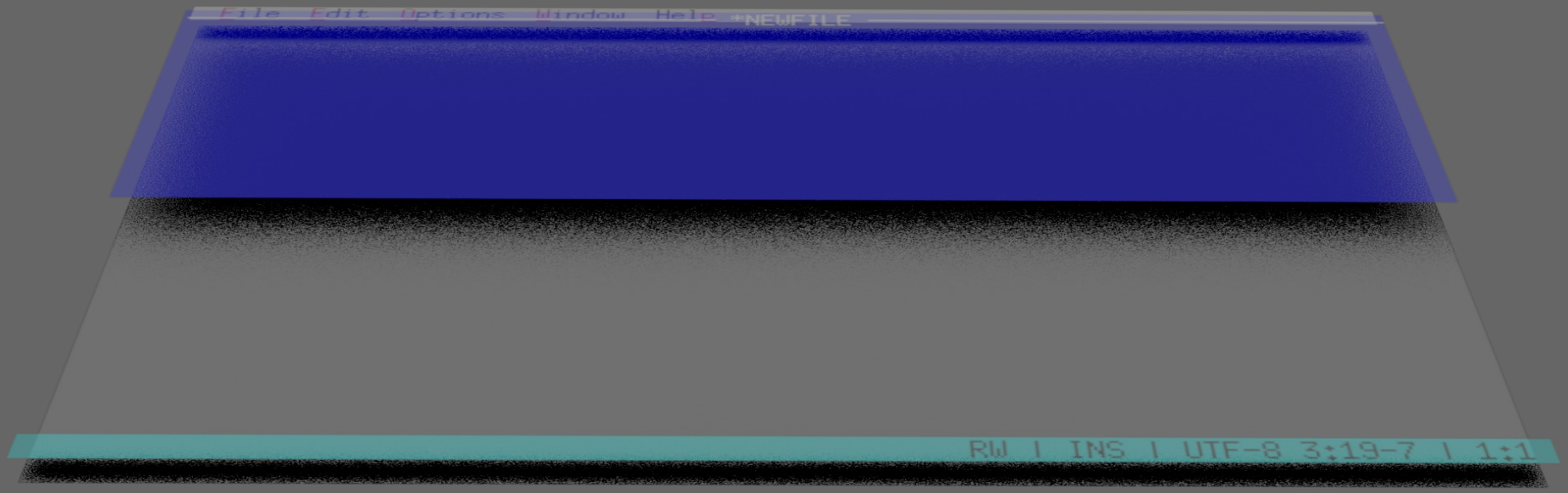

File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
>Classic
Dark
[Cancel]
[OK]
RW | INS | UTF-8 3:19-7 | 1:1

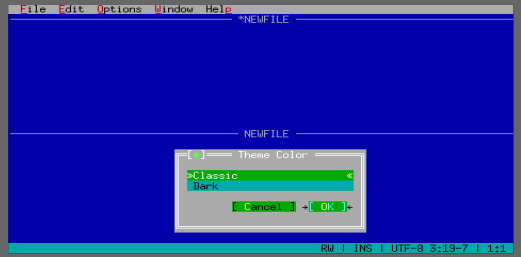
Layering


File Edit Options Window Help
RW | INS | UTF-8 3:19-7 | 1:1

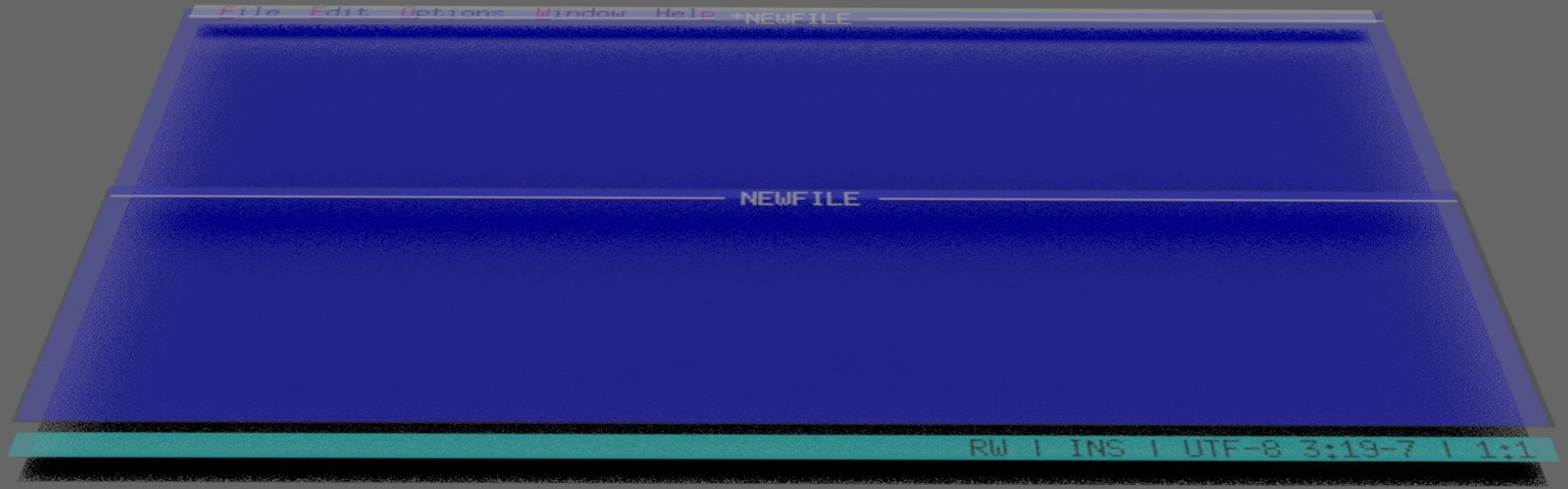

File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
>Classic
Dark
[Cancel]
[OK]
RW | INS | UTF-8 3:19-7 | 1:1

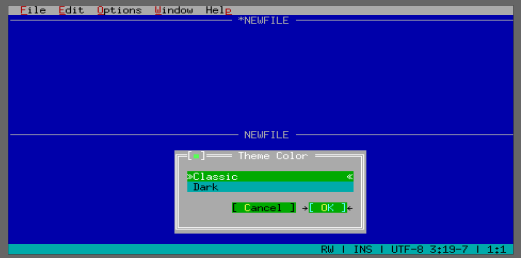
Layering


File Edit Options Window Help
*NEWFILE
RW | INS | UTF-8 3:19-7 | 1:1


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel] [OK]
RW | INS | UTF-8 3:19-7 | 1:1

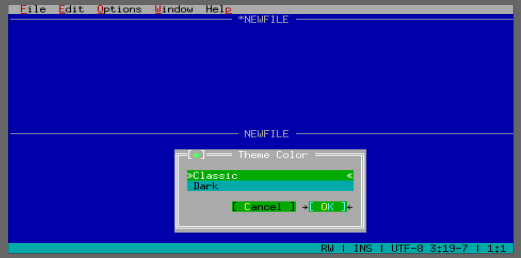
Layering


File Edit Options Window Help
*NEWFILE
NEWFILE
RW | INS | UTF-8 3:19-7 | 1:1


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
Classic
Dark
Cancel
OK
RW | INS | UTF-8 3:19-7 | 1:1

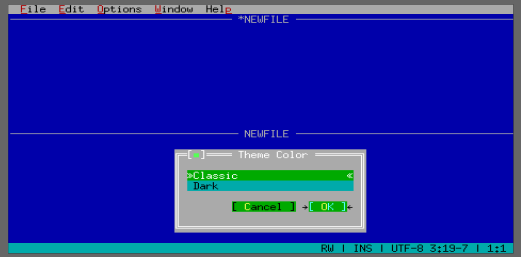
Layering


File Edit Options Window Help *NEWFILE
NEWFILE
[■]═══ Theme Color
RW | INS | UTF-8 3:19-7 | 1:1


File Edit Options Window Help
*NEWFILE
NEWFILE
[■]═══ Theme Color
»Classic «
Dark
[Cancel] [OK]
RW | INS | UTF-8 3:19-7 | 1:1

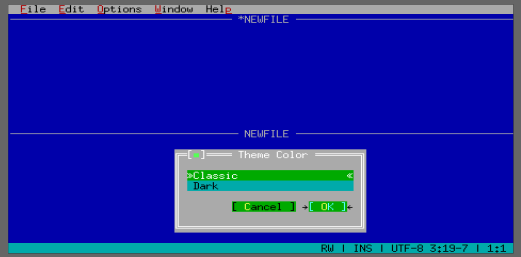
Layering


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
RW | INS | UTF-8 3:19-7 | 1:1


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel]
[OK]
RW | INS | UTF-8 3:19-7 | 1:1

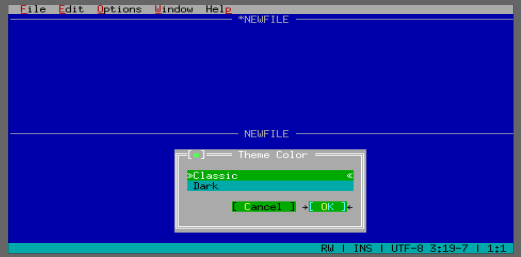
Layering


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel]
RW | INS | UTF-8 3:19-7 | 1:1


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel] [OK]
RW | INS | UTF-8 3:19-7 | 1:1

Layering


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel]
[OK]
RW | INS | UTF-8 3:19-7 | 1:1


File Edit Options Window Help
*NEWFILE
NEWFILE
Theme Color
»Classic «
Dark
[Cancel]
[OK]
RW | INS | UTF-8 3:19-7 | 1:1

Code

```
class Root : public Tui::ZRoot {
    Q_OBJECT
public:
    void terminalChanged() override;
    void quit();
};

int main(int argc, char *argv[]) {
    QCoreApplication app(argc, argv);

    Tui::ZTerminal terminal;

    Root root;

    terminal.setMainWidget(&root);

    return app.exec();
}
```

Code

```
void Root::terminalChanged() {
    // (1)
    Tui::ZShortcut *shortcut = new
Tui::ZShortcut(Tui::ZKeySequence::forKey(Tui::Key_Escape),
                                                    this, Tui::ApplicationShortcut);
    QObject::connect(shortcut,
                     &Tui::ZShortcut::activated,
                     this, &Root::quit);

    // ...
}

void Root::quit() {
    QCoreApplication::instance()->quit();
}
```

Code

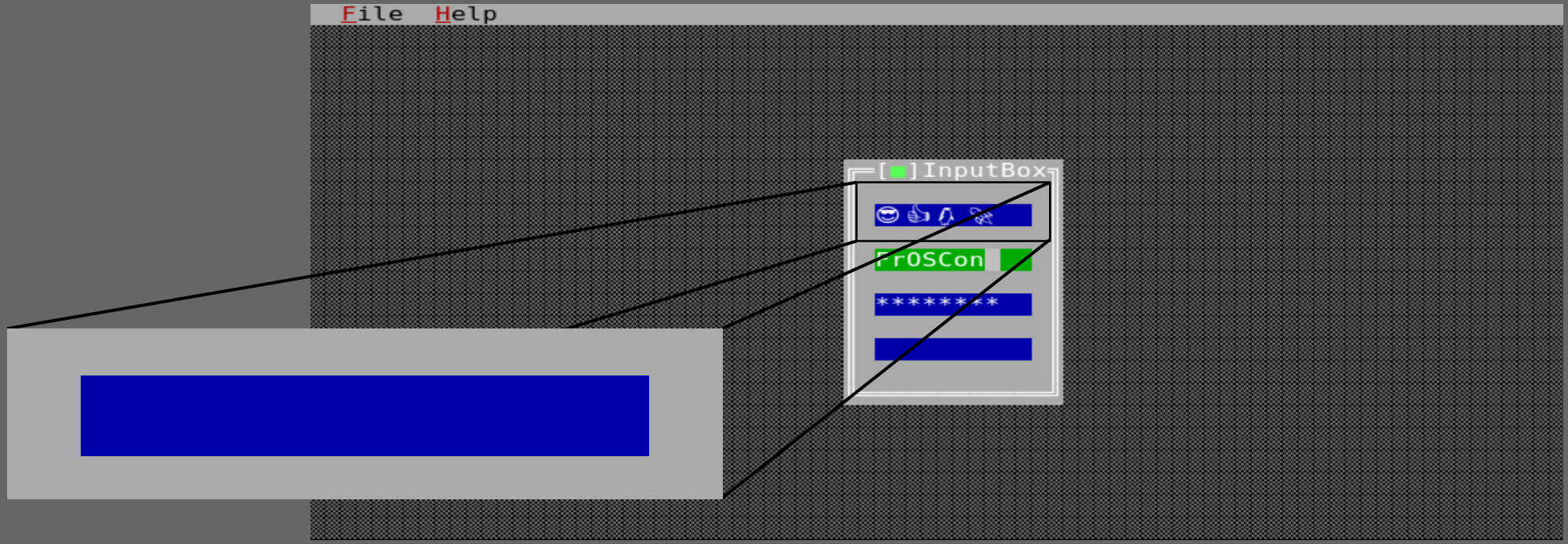
```
void Root::terminalChanged() {
    // ...

    Tui::ZWindow *win = new Tui::ZWindow("Hello World", this);
    win->setGeometry({5, 3, 20, 10});
}
```

Code

```
void Root::terminalChanged() {
    // ...
    Tui::ZButton *button = new Tui::ZButton(Tui::withMarkup, "<m>Q</m>uit", win);
    QObject::connect(button, &Tui::ZButton::clicked, this, &Root::quit);
    button->setGeometry({6, 7, 10, 1});
    button->setFocus();
}

void Root::quit() {
    QCoreApplication::instance()->quit();
}
```

InputBox

Wir wollen uns die InputBox einmal genauer anschauen:



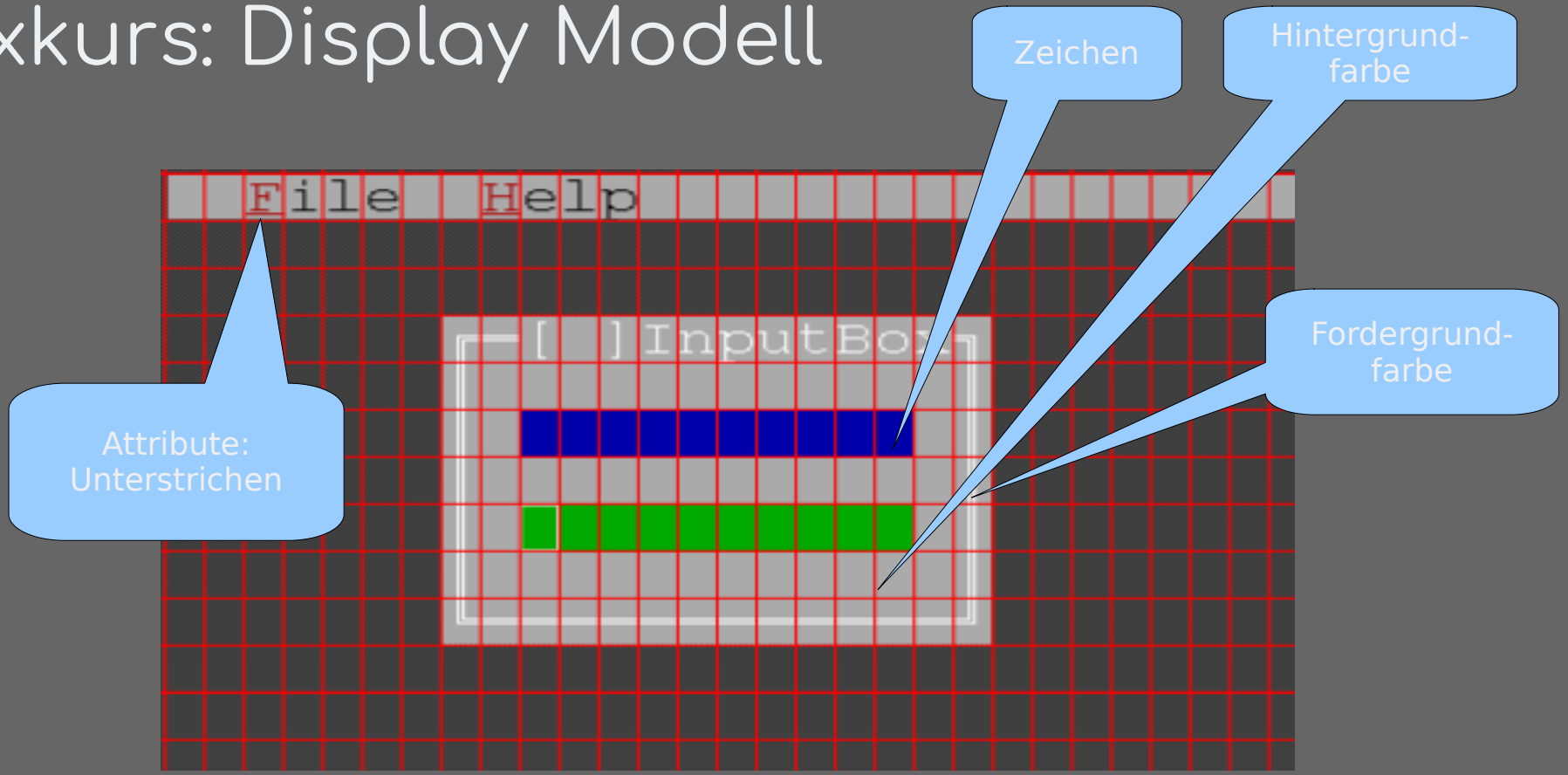
Tui::ZInputBox

„Eigentlich“ ist eine InputBox nur eine einfache Fläche...

Tui::ZInputBox

- Wir können die Fläche mit Text befüllen.

Exkurs: Display Modell



Tui::ZInputBox

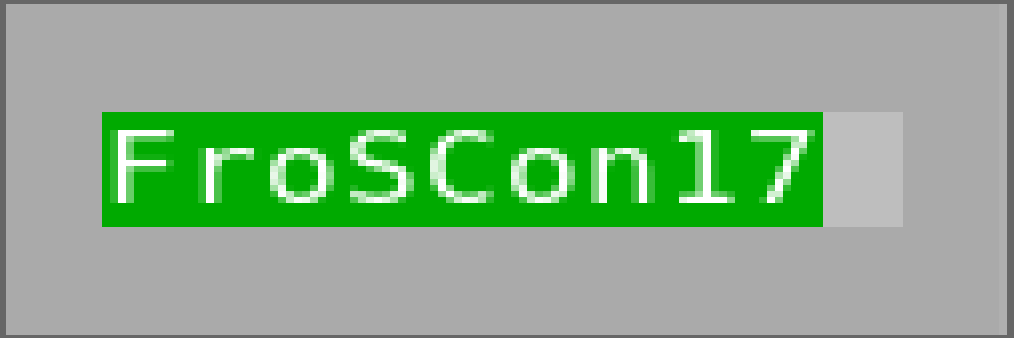
```
void ZInputBox::paintEvent(ZPaintEvent *event) {
    QRect r = contentsRect();
    ZPainter painter = event->painter()->translateAndClip(r.left(), r.top(),
                                                           r.width(), r.height());
    ZTextLayout textlayout = getTextLayoutForPaint();

    ZColor bg = ...;
    ZColor fg = ...;

    painter.clear(fg, bg);
    textlayout.draw(painter, {0, 0}, {fg, bg});
}
```

Tui::ZInputBox

- Wir brauchen einen Cursor

Tui::ZInputBox

- Wir brauchen einen Cursor




```
textlayout.draw(painter, {-p->scrollPosition, 0}, {fg, bg});
textlayout.showCursor(painter, {-p->scrollPosition, 0}, p->cursorPosition);
```

Tui::ZInputBox

- keyEvent
  Backspace, Delete, Left, Right, Home und Ende

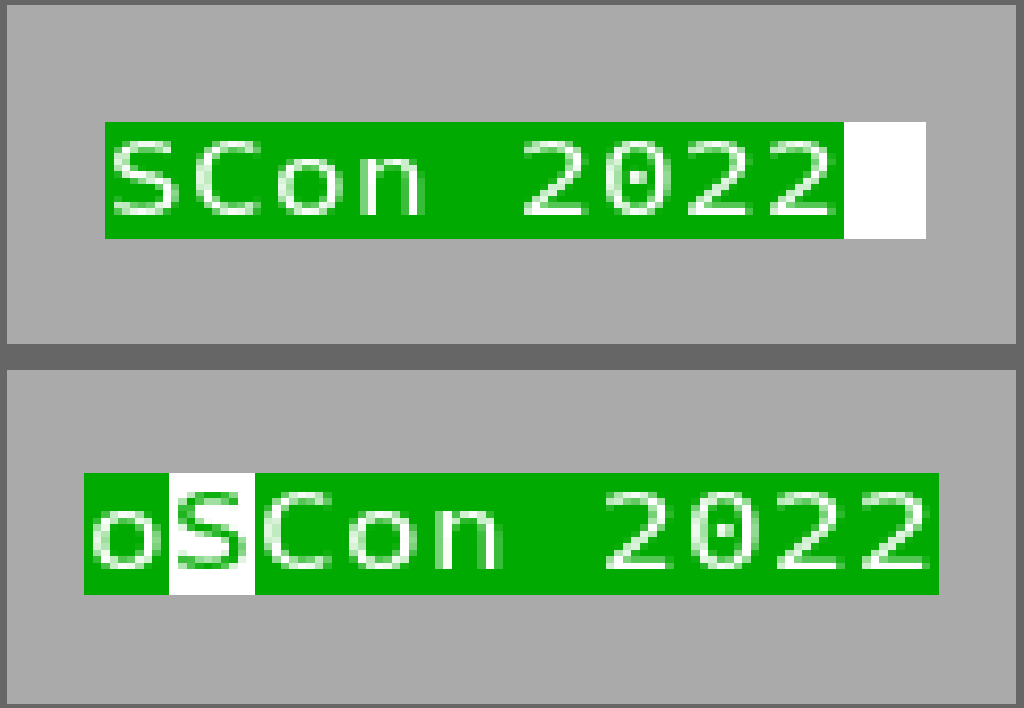
Tui::ZInputBox

Scrolling


```
textlayout.draw(painter, {-p->scrollPosition, 0}, {fg, bg});
textlayout.showCursor(painter, {-p->scrollPosition, 0}, p->cursorPosition);
```

Tui::ZInputBox

Immer ein Zeichen im Umfeld des Cursors sichtbar:



Tui::ZInputBox

- Scroling nur von ganzen Zeichen
- Zeichen aus mehreren Code Units

Tui::ZInputBox

- A + U+0325 COMBINING RING BELOW

Tui::ZInputBox

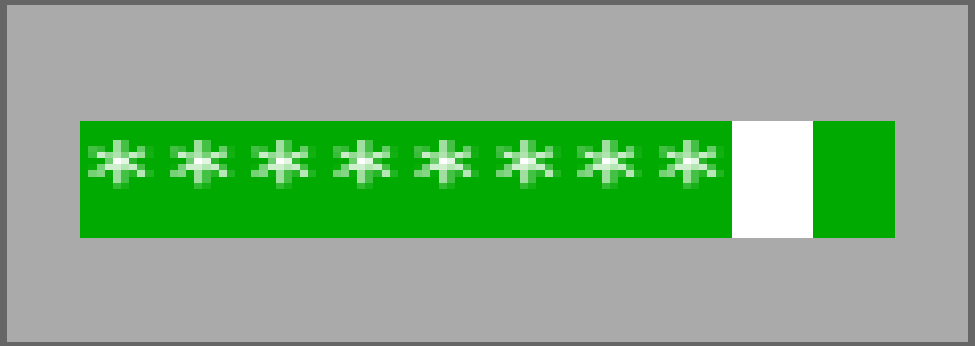
- Paste
  - Getrennt von Tasteneingabe
- Kein Mouse Modus (wip)

Tui::ZInputBox

```
void ZInputBox::paintEvent(ZPaintEvent *event) {
    QRect r = contentsRect();
    ZPainter painter = event->painter()->translateAndClip(r.left(), r.top(),
                                                          r.width(), r.height());
    ZTextLayout textlayout = ZTextLayout(terminal()->textMetrics(), _text);
    textlayout.doLayout(65000);

    ZColor bg = ..., fg = ...;

    painter.clear(fg, bg);
    if (focus()) {
        textlayout.draw(painter, {-p->scrollPosition, 0}, {fg, bg});
        textlayout.showCursor(painter, {-p->scrollPosition, 0},
                              p->cursorPosition);
    } else {
        textlayout.draw(painter, {0, 0}, {fg, bg});
    }
}
```

Tui::ZInputBox

- Passwortfelder

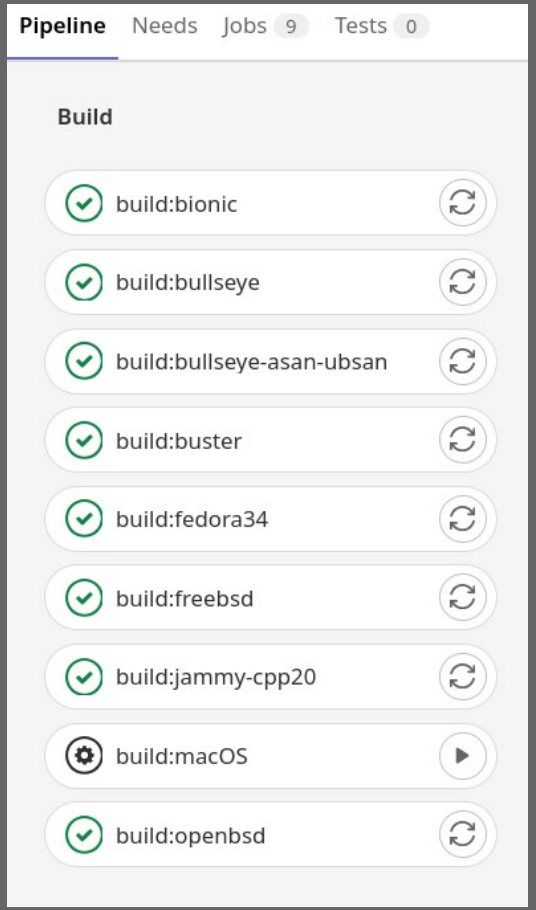


Intuitif

- ESC schließt Dialoge.
- TAB zum nächsten Element wechseln.
- F6 oder Shift + F6 Fenster wechseln.
- ALT + - öffnet das Fenster/Dialog Menu. []
- ALT + a-z0-9 wählt nach Kürzel aus. Durch Farbe oder Unterstreichung gekennzeichnet.
- Ctrl + a-z Tastenkürzel für weitere Aktionen.

Testing & Software Qualität

- Natürlich CI
  - soll auch mit älteren Distros laufen
  - und auch auf BSDs
- Tests von getrennten Komponenten
  - möglichst nur eine Klasse die getestet wird.
- Größtenteils über öffentliche API
  - zur Not Tests über private Schnitstellen
- Always more to test



Testing & Software Qualität

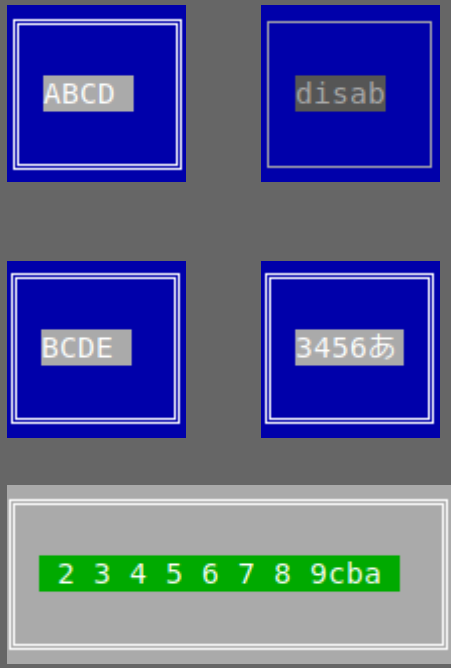
- Für Widgets “visuelle Tests”
- Testet die Ausgabe
  - .tpi Format, nicht Pixel

.tpi

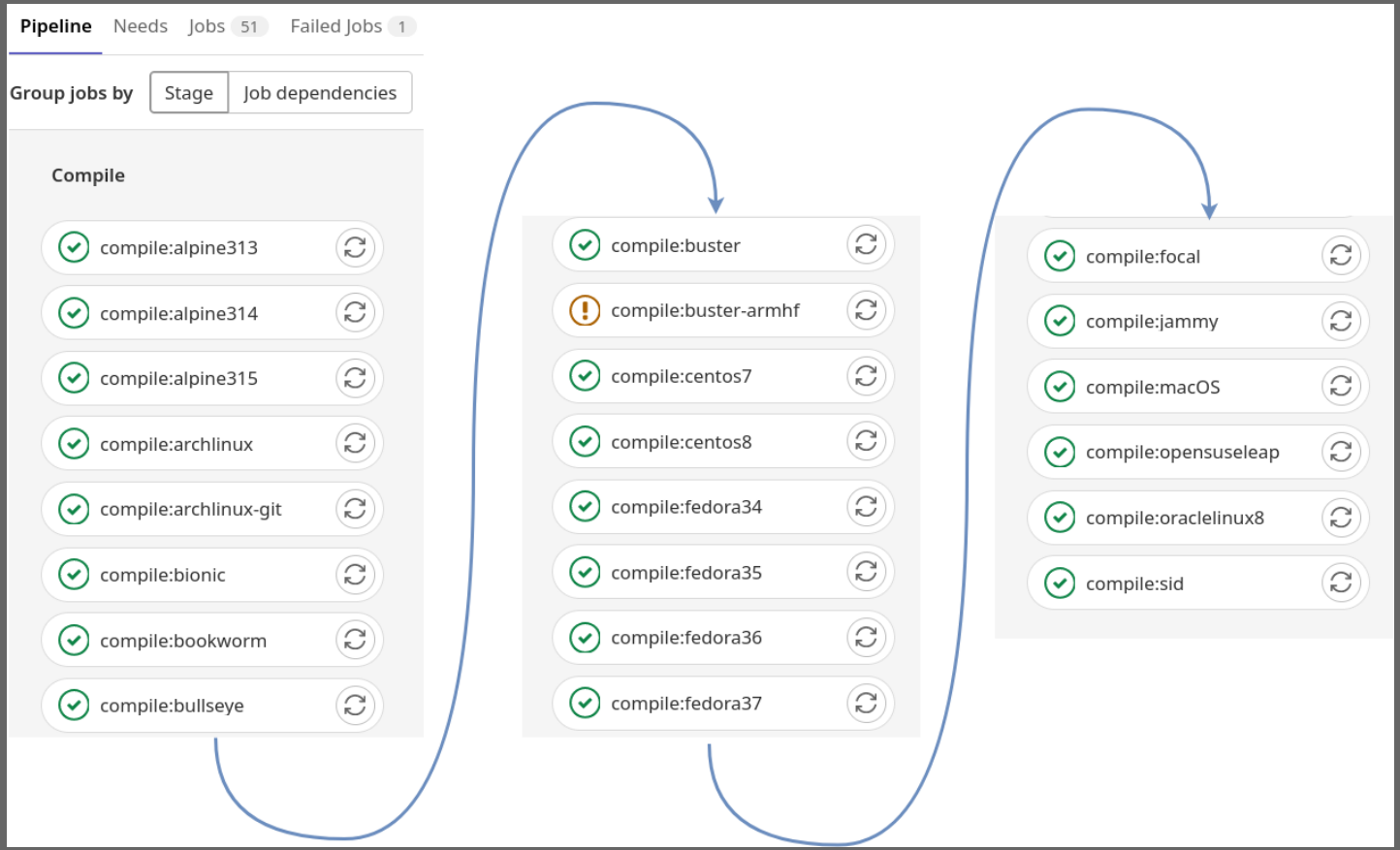

```
{"termpaint_image": true,
  "width": 10, "height": 5, "version": 0, "cells":[
    {"x": 0, "y": 0,
     "t": "\u2554", "fg": "#ffffff", "bg": "#0000aa"},
    {"x": 1, "y": 0,
     "t": "\u2550", "fg": "#ffffff", "bg": "#0000aa"},
    {"x": 2, "y": 0,
     "t": "\u2550", "fg": "#ffffff", "bg": "#0000aa"},
```

Testing & Software Qualität

- Zum Teil auch für Logik
  die über public API nicht testbar ist
  z.B. scrolling
- Für Paletten support
- für ZInputBox über 100 Stück



Mehr CI


Pipeline
Needs
Jobs 51
Failed Jobs 1
Group jobs by
Stage
Job dependencies
Compile
compile:alpine313
compile:alpine314
compile:alpine315
compile:archlinux
compile:archlinux-git
compile:bionic
compile:bookworm
compile:bullseye
compile:buster
compile:buster-armhf
compile:centos7
compile:centos8
compile:fedora34
compile:fedora35
compile:fedora36
compile:fedora37
compile:focal
compile:jammy
compile:macOS
compile:opensuseleap
compile:oraclelinux8
compile:sid

Stand

- Work in Progress
  - breaking changes in Details
- Grundlagen und einige Widgets
- Ausführliche Doku fehlt -> Header anschauen
- Details noch im Fluss
- Neue Widgets in Planung:
  - Multiline Text Edit, Treeview, TableView, MessageBox, ...
- Offen für Feedback und Mithilfe
  - gerne auch Pullrequests

Questions?

https://github.com/tuiwidgets/tuiwidgets
https://tuiwidgets.namepad.de

CONTACT:

Christoph Hüffelmann <chr@istoph.de>
Martin Hostettler <textshell@uchuujin.de>