FOSDEM 2022

Back to DirectFB!

The revival of DirectFB with DirectFB2

Nicolas Caramelli

20 years of history

2001 DirectFB-0.9.0, the initial public release

"X is dead" is announced on *directfb.org*

→ This official site has disappeared for a few years ...

2007 1st release of the stable 1.0 series, 1.1 dev branch

2008 1st release of the stable 1.2 series, 1.3 dev branch

2009 1st release of the stable 1.4 series, 1.5 dev branch

2012 1st release of the stable 1.6 series, 1.7 dev branch

2015 1.7.7 is the last official DirectFB release

2016 Last commit for the never published 1.8 series

→ https://github.com/deniskropp/DirectFB

20 years of history

DirectFB eventually died before X ...

Or so it seemed until DirectFB2, a fork of DirectFB, was created in order to maintain DirectFB for its use on embedded systems.

➡ This is not a replacement for window system such as X or Wayland, but an option that primarily targets small systems !

2021 The initial public version of DirectFB2

➡ https://github.com/directfb2/DirectFB2

FOSDEM 2022

Contents

Contents

DirectFB2 overview

- Switch on the **Meson** build system
- **Pure C** implementation
- **No external dependency**, except:
  - libc (glibc, uClibc, musl, ...)
  - fluxcomp, a tool for converting .flux interface description files to .c files (only required on Host for the building)

    ➡ https://github.com/deniskropp/flux
- **Linux** support only at this time, but support for other embedded OS expected
- **Modularization** of the source code

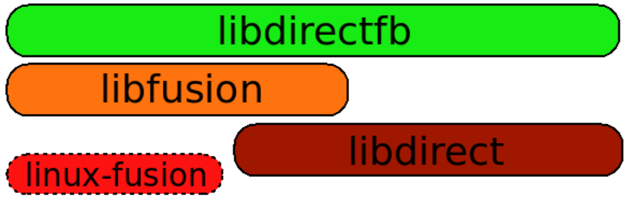
  ➡ splitting of the original DirectFB repository

DirectFB2 core repository

- DirectFB API (interfaces for EventBuffer, Surface, Window, Font, Image, Video, ...)

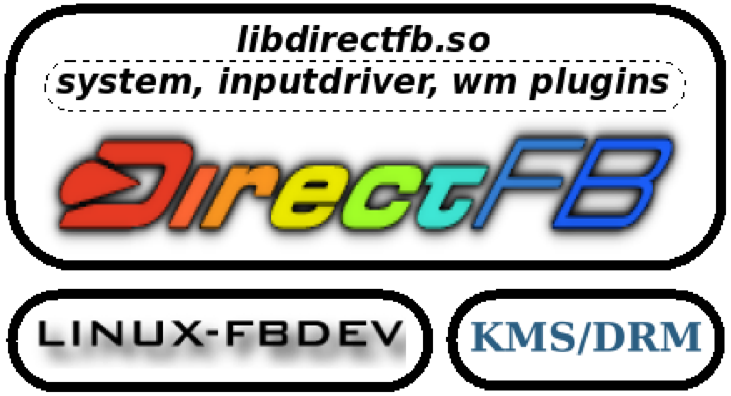
  ➡ *include/directfb.h* header (backward compatible with original API)

- libdirectfb library in *src/* directory based on 2 internal libs



  - libdirect low-level library
  - libfusion IPC library with optional linux-fusion kernel module which implements the critical parts of Fusion

- generic system module for supported OS (currently Linux)
  - legacy FBDev
  - modern DRM/KMS

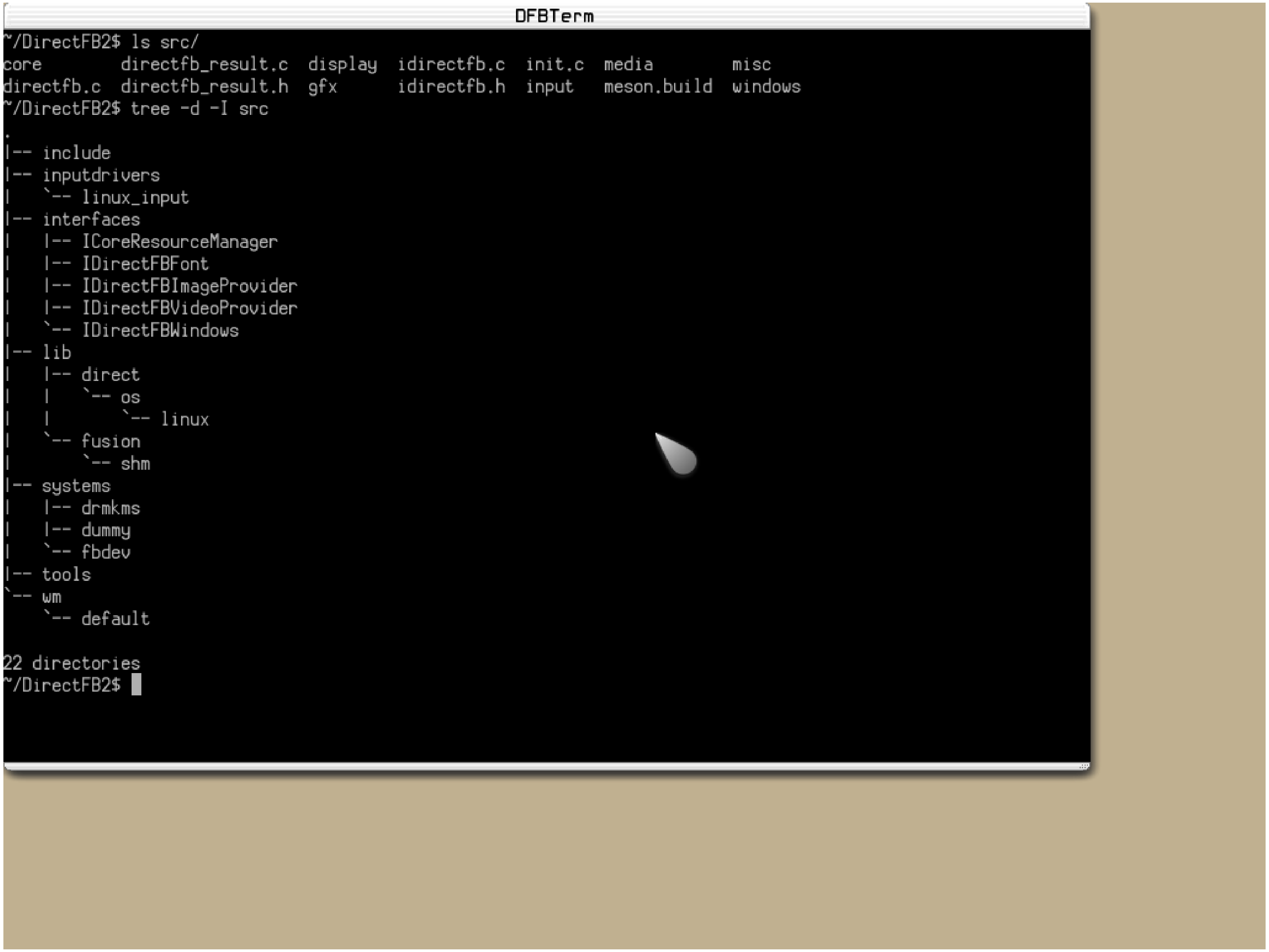


- generic input driver module

  Linux input driver

- default WM module (window manager)

DirectFB2 core repository



DFBTerm

```
~/DirectFB2$ ls src/
core        directfb_result.c  display  idirectfb.c  init.c  media       misc
directfb.c  directfb_result.h  gfx      idirectfb.h  input   meson.build windows
~/DirectFB2$ tree -d -I src
.
|-- include
|-- inputdrivers
|   `-- linux_input
|-- interfaces
|   |-- ICoreResourceManager
|   |-- IDirectFBFont
|   |-- IDirectFBImageProvider
|   |-- IDirectFBVideoProvider
|   `-- IDirectFBWindows
|-- lib
|   |-- direct
|   |   `-- os
|   |       `-- linux
|   `-- fusion
|       `-- shm
|-- systems
|   |-- drmkms
|   |-- dummy
|   `-- fbdev
|-- tools
`-- wm
    `-- default

22 directories
~/DirectFB2$ 
```

DirectFB2 core repository

- The *interfaces/* directory in the DirectFB2 core repository contains:
  - the DGIFF (*DirectFB Glyph Image File Format*) font provider
  - the DFIFF (*DirectFB Fast Image File Format*) image provider
  - the DFVFF (*DirectFB Fast Video File Format*) video provider

→ These basic providers allow rendering of **raw font, raw image or raw video** (without any dependencies)

- *tools/* directory
  - dfbg to configure the background
  - dfbinfo to print DirectFB settings

Not in the DirectFB2 core repository

Separate repositories from the DirectFB2 core are used for:

- Additional system and input driver modules
- Additional WM modules (like SaWMan)
- Additional font/image/video providers
- GFX driver modules (chipset hardware acceleration)

  → HW acceleration of graphics operations such as blitting, rectangle/triangle/line drawing, blending, color keying ...

- DiVine (DirectFB Virtual input extension)
- FusionSound (audio subsystem using Fusion IPC)
- ...

Contents

DirectFB2 install

- Configuration
  - single application core (default build configuration)

    ➡ one application can be running

    ```
    $ meson build/
    ```
  - multi application core

    ➡ multiple applications to run at the same time

    1. with Fusion implemented completely in user space

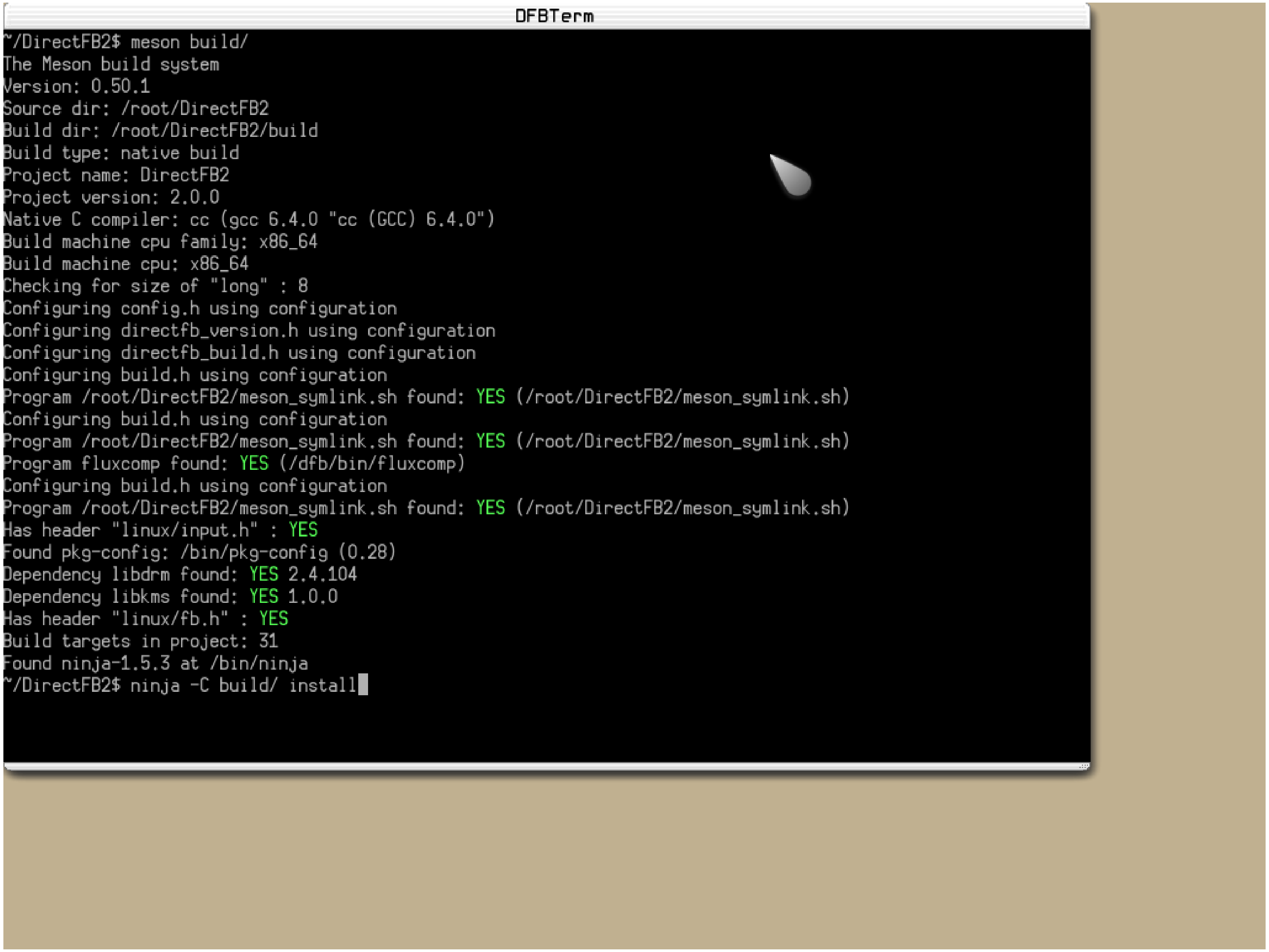
       ```
       $ meson -Dmulti=true build/
       ```
    2. with Fusion based on the linux-fusion kernel module

       ```
       $ meson -Dmulti=true -Dmulti-kernel=true build/
       ```
- Build / Install

  ```
  $ ninja -C build/
  $ ninja -C build/ install
  ```

FOSDEM 2022

DirectFB2 install



DFBTerm

```
~/DirectFB2$ meson build/
The Meson build system
Version: 0.50.1
Source dir: /root/DirectFB2
Build dir: /root/DirectFB2/build
Build type: native build
Project name: DirectFB2
Project version: 2.0.0
Native C compiler: cc (gcc 6.4.0 "cc (GCC) 6.4.0")
Build machine cpu family: x86_64
Build machine cpu: x86_64
Checking for size of "long" : 8
Configuring config.h using configuration
Configuring directfb_version.h using configuration
Configuring directfb_build.h using configuration
Configuring build.h using configuration
Program /root/DirectFB2/meson_symlink.sh found: YES (/root/DirectFB2/meson_symlink.sh)
Configuring build.h using configuration
Program /root/DirectFB2/meson_symlink.sh found: YES (/root/DirectFB2/meson_symlink.sh)
Program fluxcomp found: YES (/dfb/bin/fluxcomp)
Configuring build.h using configuration
Program /root/DirectFB2/meson_symlink.sh found: YES (/root/DirectFB2/meson_symlink.sh)
Has header "linux/input.h" : YES
Found pkg-config: /bin/pkg-config (0.28)
Dependency libdrm found: YES 2.4.104
Dependency libkms found: YES 1.0.0
Has header "linux/fb.h" : YES
Build targets in project: 31
Found ninja-1.5.3 at /bin/ninja
~/DirectFB2$ ninja -C build/ install
```

DirectFB2 cross-compilation

- Example for an ARM target ➡ create the *arm-linux-gnueabihf* cross file

```
[binaries]
c = 'arm-linux-gnueabihf-gcc'
strip = 'arm-linux-gnueabihf-strip'
pkgconfig = 'pkg-config'

[host_machine]
system = 'linux'
cpu_family = 'arm'
cpu = 'armv7-a'
endian = 'little'
```

➡
```
$ meson --cross-file arm-linux-gnueabihf build/
```

- Library size around 1M

Contents

DirectFB-examples

- DirectFB demos and test programs

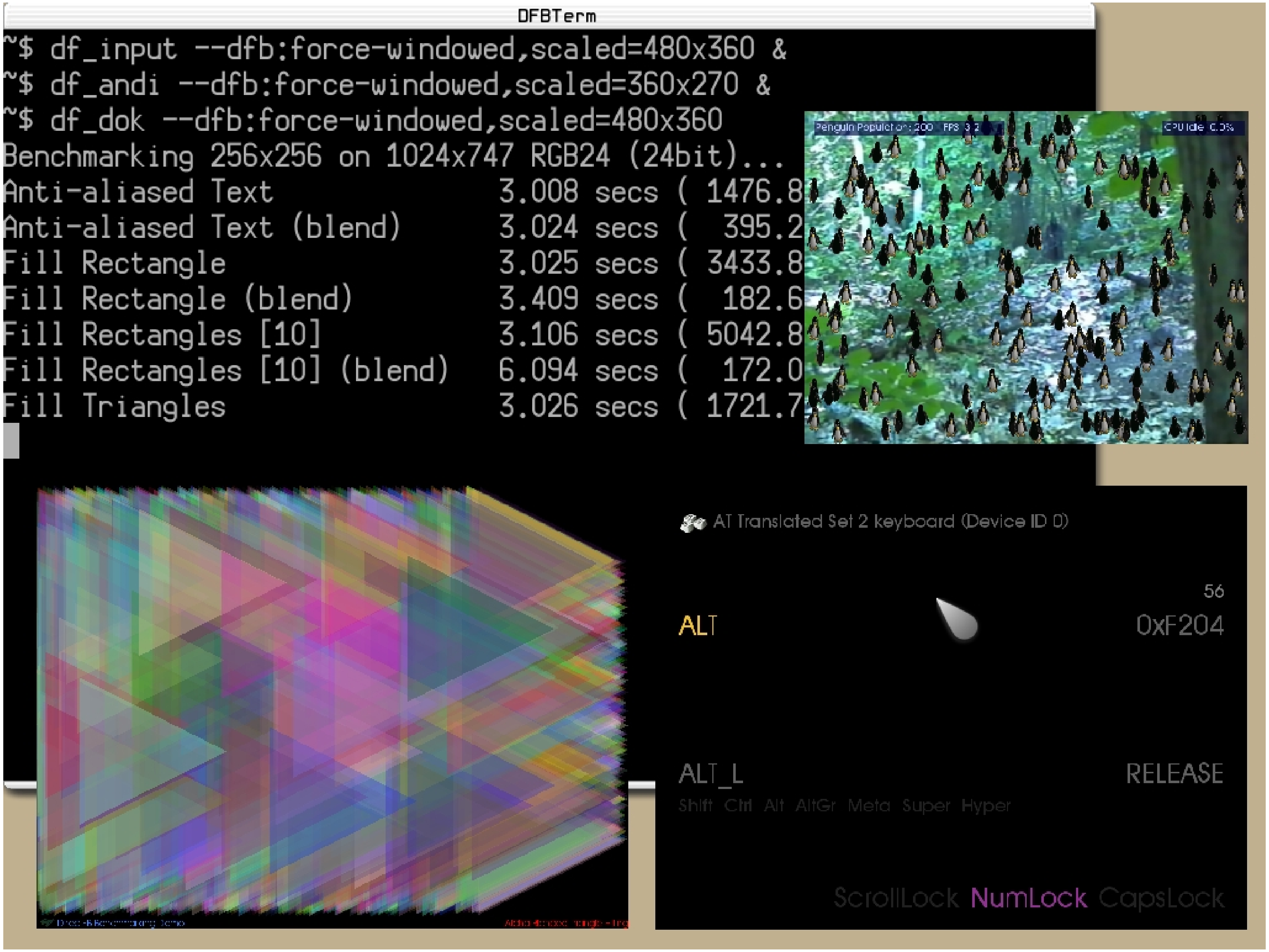
  ➡ https://github.com/directfb2/DirectFB-examples

• df_andi	penguin animation
• df_dok	benchmarking program
• df_fire	fire effect demo
• df_input	test application for input devices
• df_knuckles	3D skull drawn using triangles
• df_matrix	transformation matrix example
• df_neo	scaling animation with alpha blending / color modulation
• df_particle	moving fountain demo
• df_texture	texture mapping example
• df_video	video playback in a moving window
• df_window	window stack example
• …	

- Compared to the latest DirectFB-examples-1.7.0 released in 2013
  - ➔ Like the DirectFB2 core repository, the examples are built using **Meson**
  - ➔ The examples now only use the basic DGIFF / DFIFF/ DFVFF providers

FOSDEM 2022

DirectFB-examples



DFBTerm

```
~$ df_input --dfb:force-windowed,scaled=480x360 &
~$ df_andi --dfb:force-windowed,scaled=360x270 &
~$ df_dok --dfb:force-windowed,scaled=480x360
Benchmarking 256x256 on 1024x747 RGB24 (24bit)...
Anti-aliased Text            3.008 secs ( 1476.8
Anti-aliased Text (blend)    3.024 secs (  395.2
Fill Rectangle               3.025 secs ( 3433.8
Fill Rectangle (blend)       3.409 secs (  182.6
Fill Rectangles [10]         3.106 secs ( 5042.8
Fill Rectangles [10] (blend) 6.094 secs (  172.0
Fill Triangles               3.026 secs ( 1721.7
```

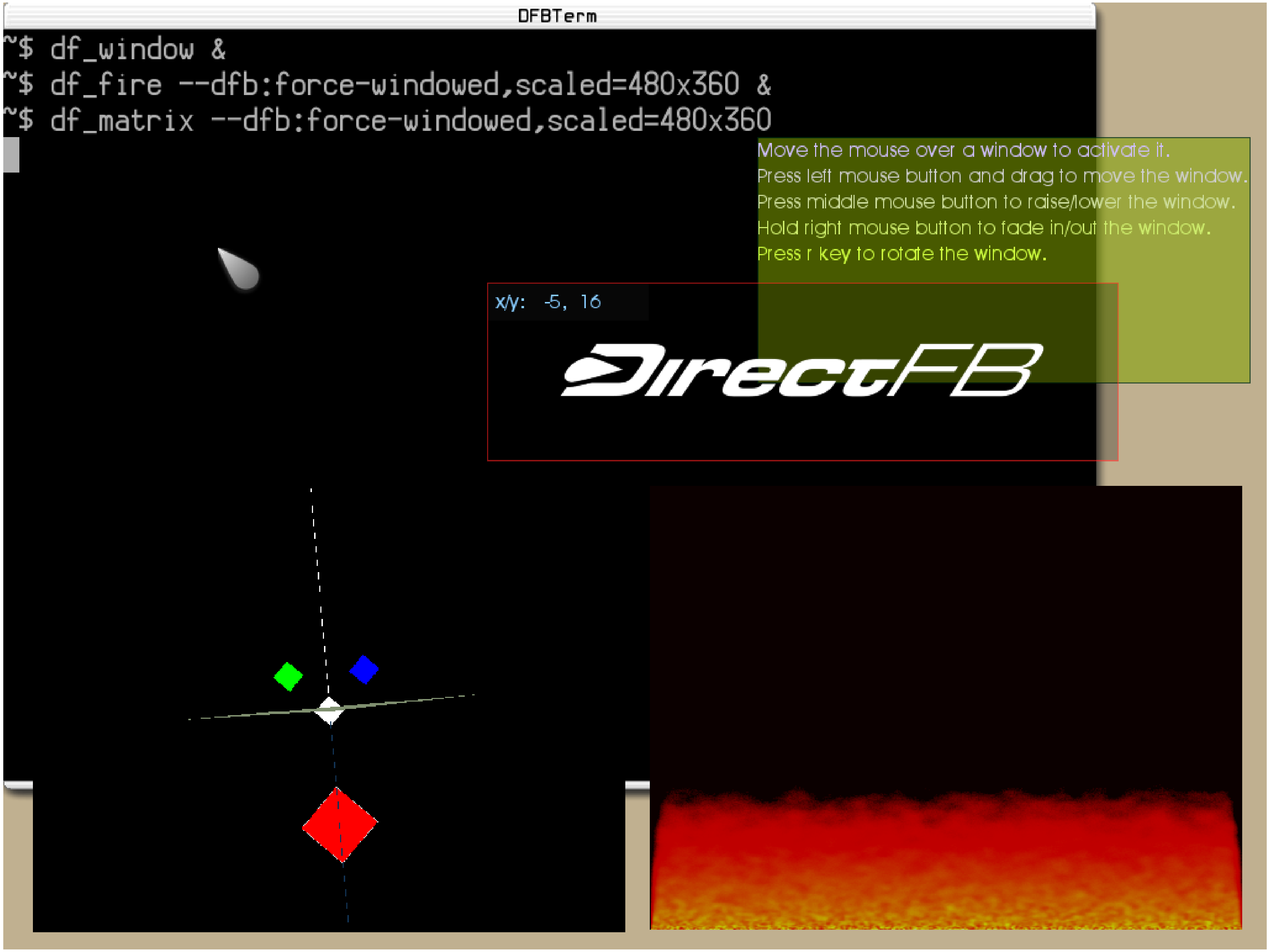
AT Translated Set 2 keyboard (Device ID 0)

```
                                        56
ALT                                 0xF204

ALT_L                              RELEASE
Shift Ctrl Alt AltGr Meta Super Hyper

          ScrollLock NumLock CapsLock
```

DirectFB-examples

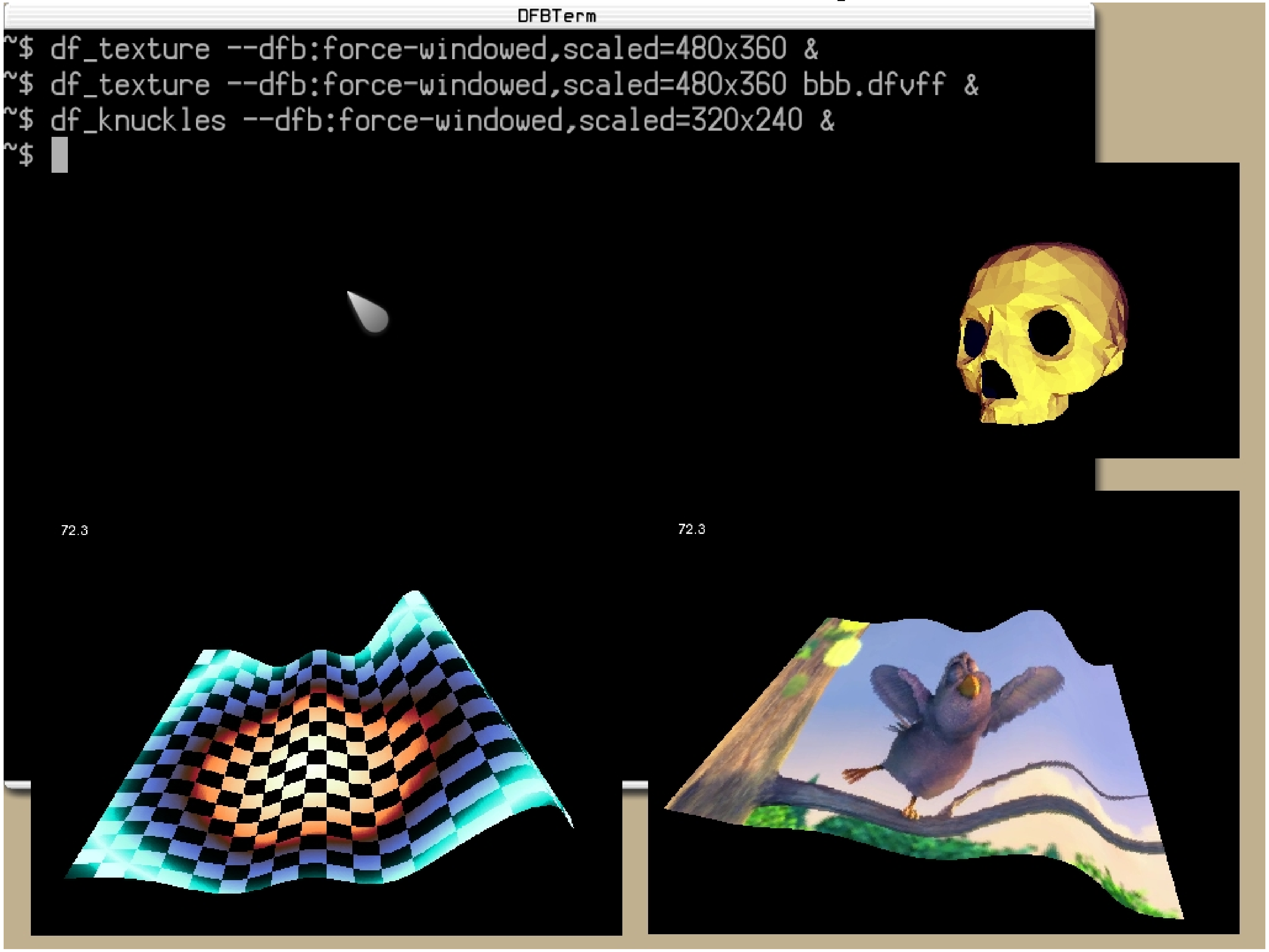


```
~$ df_window &
~$ df_fire --dfb:force-windowed,scaled=480x360 &
~$ df_matrix --dfb:force-windowed,scaled=480x360
```

Move the mouse over a window to activate it.
Press left mouse button and drag to move the window.
Press middle mouse button to raise/lower the window.
Hold right mouse button to fade in/out the window.
Press r key to rotate the window.

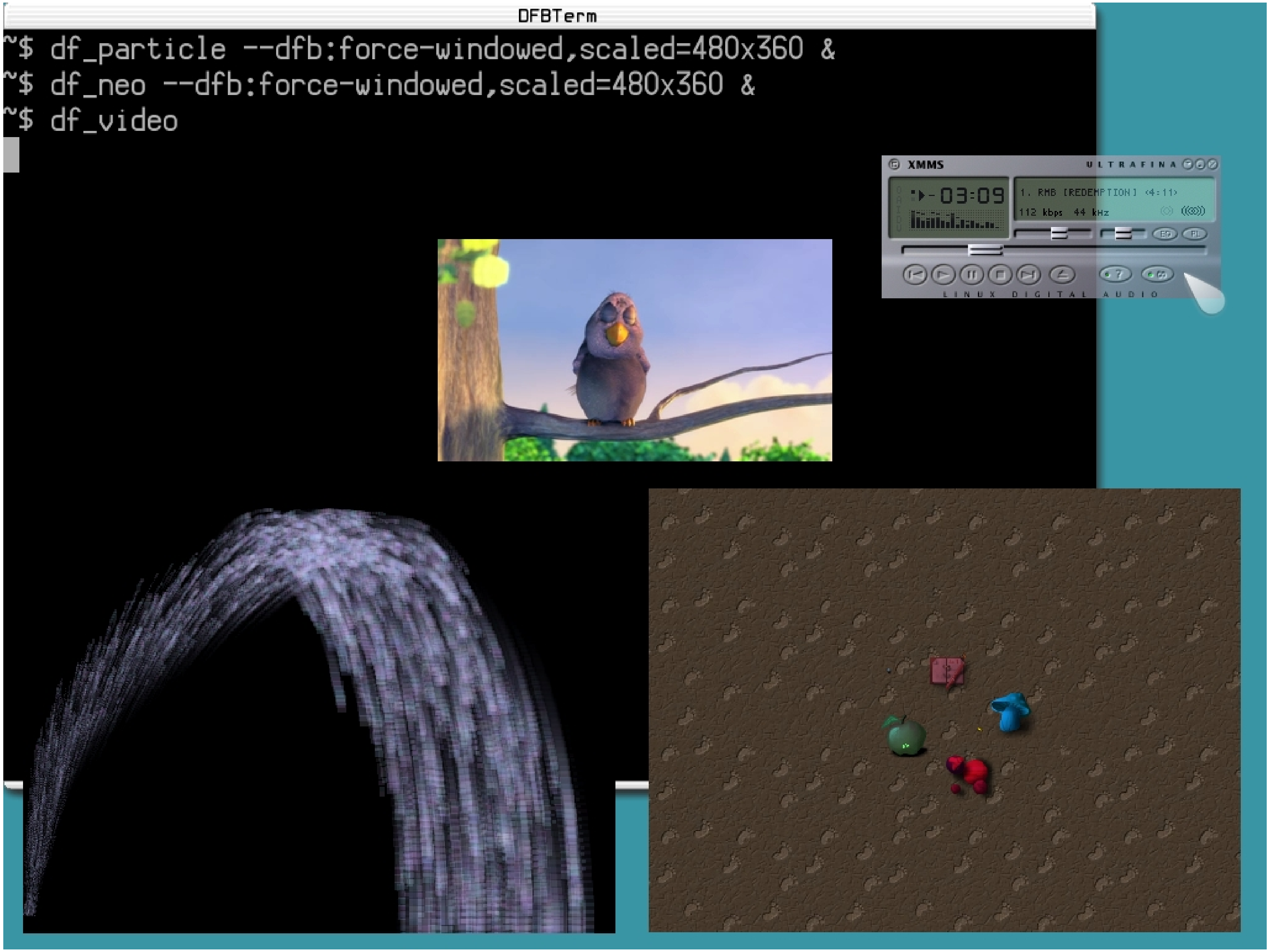
x/y: -5, 16

DirectFB-examples



```
~$ df_texture --dfb:force-windowed,scaled=480x360 &
~$ df_texture --dfb:force-windowed,scaled=480x360 bbb.dfvff &
~$ df_knuckles --dfb:force-windowed,scaled=320x240 &
~$ 
```

DirectFB-examples



```
~$ df_particle --dfb:force-windowed,scaled=480x360 &
~$ df_neo --dfb:force-windowed,scaled=480x360 &
~$ df_video
```

Contents

DirectFB2-media

- Additional font/image/video providers, coming from the original DirectFB

  → https://github.com/directfb2/DirectFB2-media

• FreeType2	font provider based on *freetype.org*
• PNG	image provider based on *libpng.org*
• JPEG	image provider based on ijg.org or *libjpeg-turbo.org*
• FFmpeg	video provider based on *ffmpeg.org*
• GStreamer	video provider based on *gstreamer.freedesktop.org*
• …	

  → complements the basic DGIFF / DFIFF / DFVFF providers and depends on external libraries

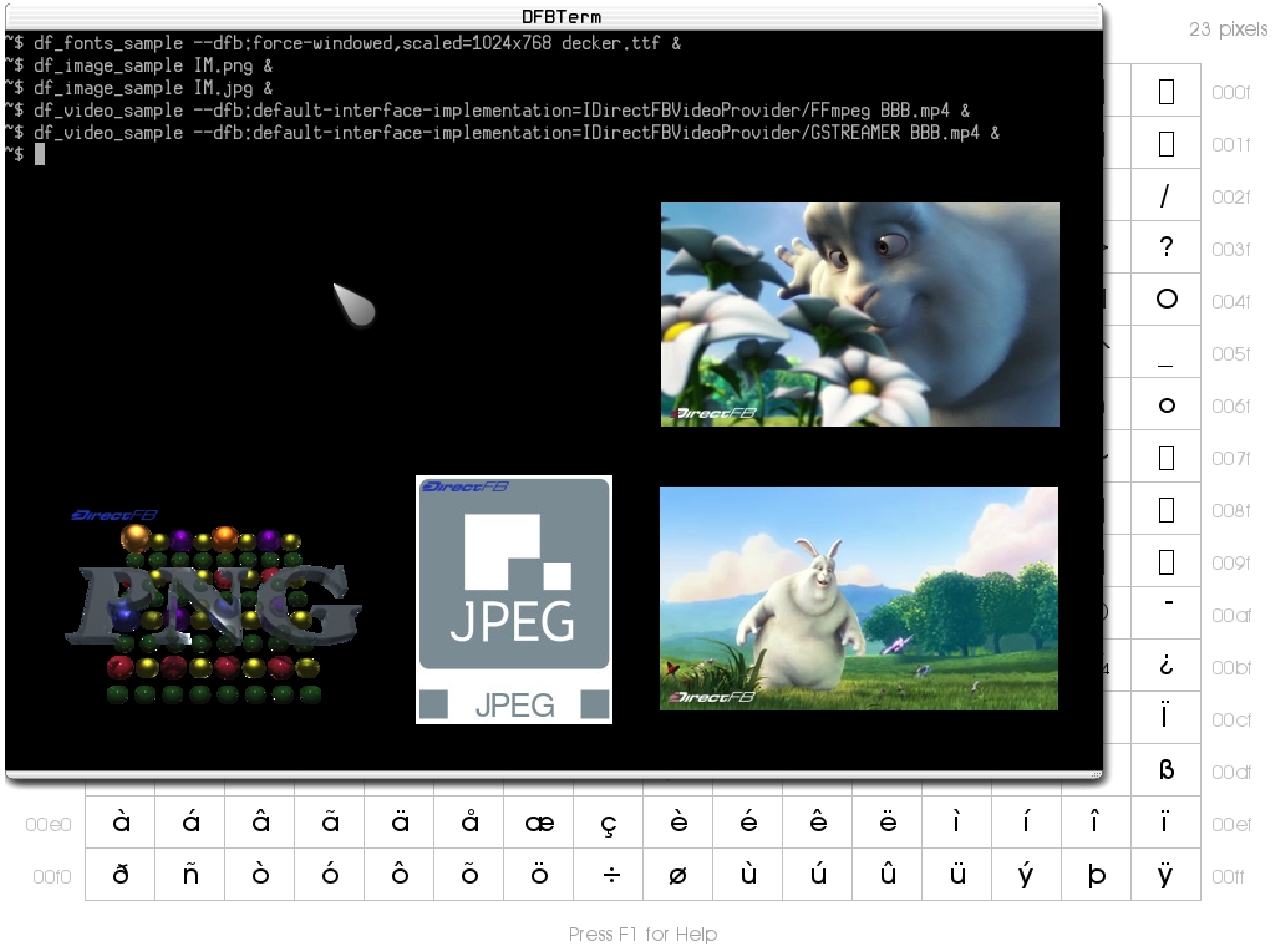
- Providers are probed by DirectFB for finding a suitable provider

  → note that if 2 providers can handle a media, it is always possible to probe one first with option ***--dfb:default-interface-implementation***

- df_fonts_sample / df_image_sample / df_video_sample viewers

FOSDEM 2022

DirectFB2-media



DFBTerm

```
~$ df_fonts_sample --dfb:force-windowed,scaled=1024x768 decker.ttf &
~$ df_image_sample IM.png &
~$ df_image_sample IM.jpg &
~$ df_video_sample --dfb:default-interface-implementation=IDirectFBVideoProvider/FFmpeg BBB.mp4 &
~$ df_video_sample --dfb:default-interface-implementation=IDirectFBVideoProvider/GSTREAMER BBB.mp4 &
~$ █
```

▯	000f
▯	001f
/	002f
?	003f
O	004f
_	005f
o	006f
▯	007f
▯	008f
▯	009f
-	00af
¿	00bf
Ï	00cf
ß	00df

00e0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	00ef
00f0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	00ff

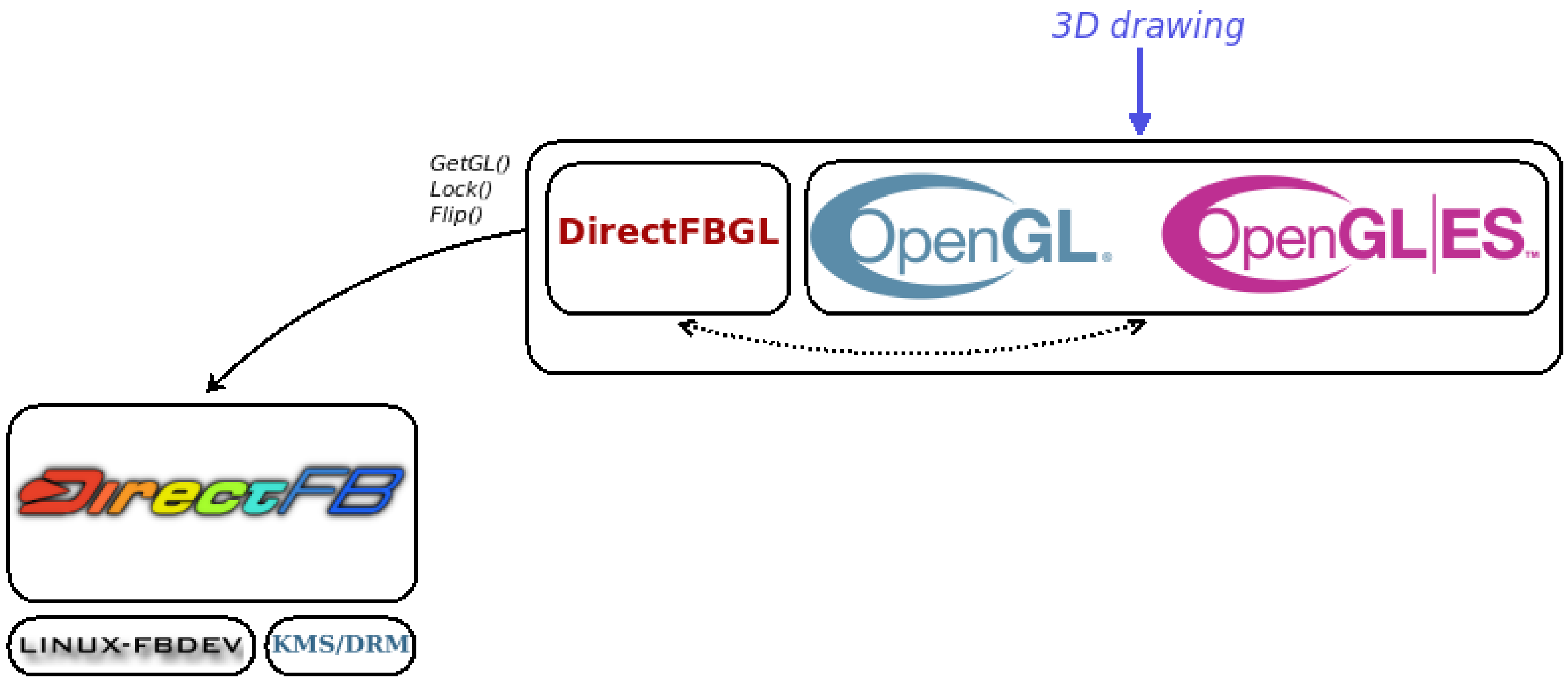
Contents

OpenGL rendering

- Applications can choose between 2 interfaces for rendering:
  - **DirectFBGL** (OpenGL extension for DirectFB)
  - **EGL** for the DirectFB platform
- The **Mesa 3D** project makes OpenGL and OpenGL ES rendering possible with DirectFB for these 2 interfaces

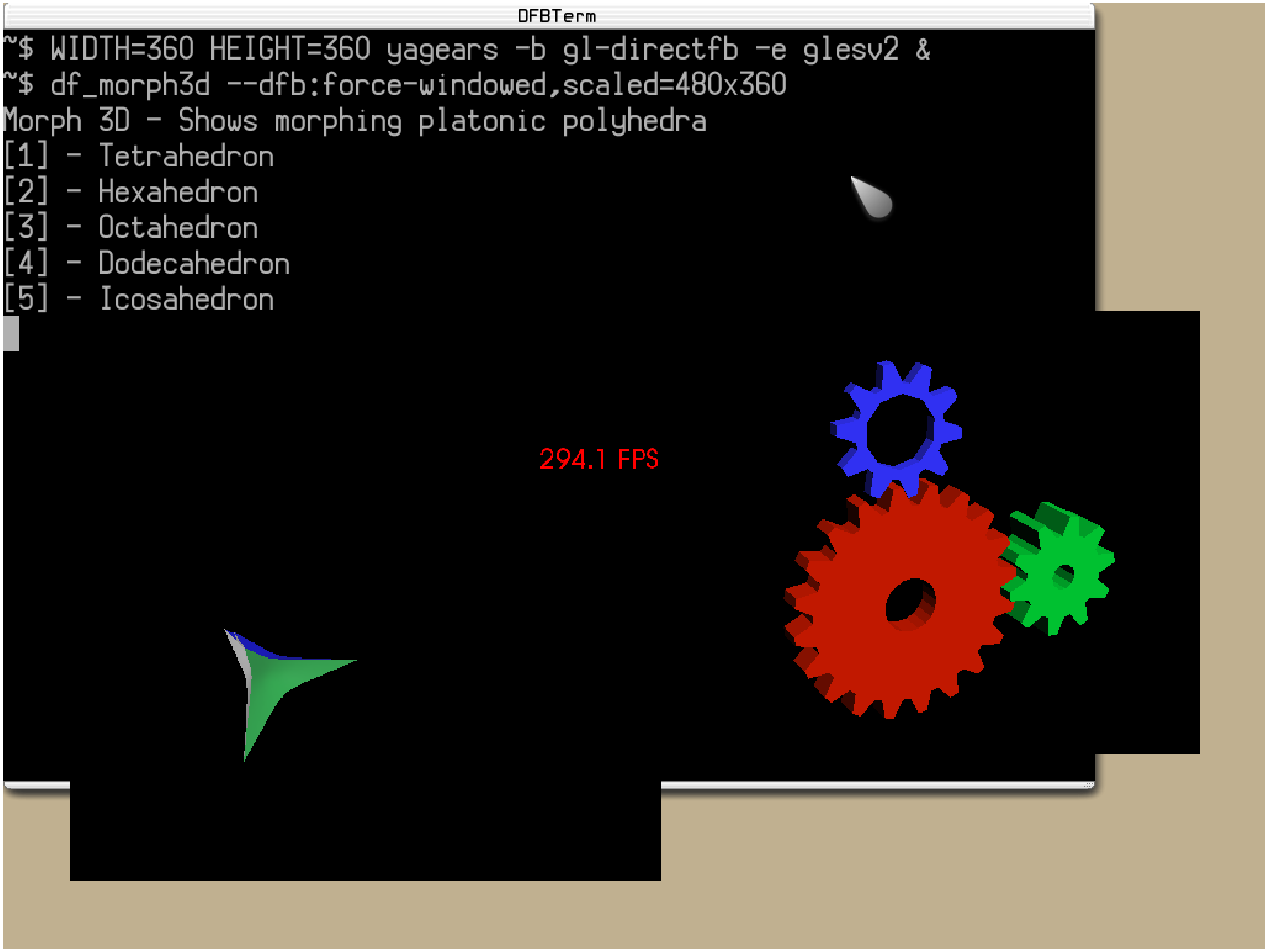
➡ mainly used for experimentation and debugging purposes, depending on the chipset, a specific implementation may be available

DirectFBGL interface



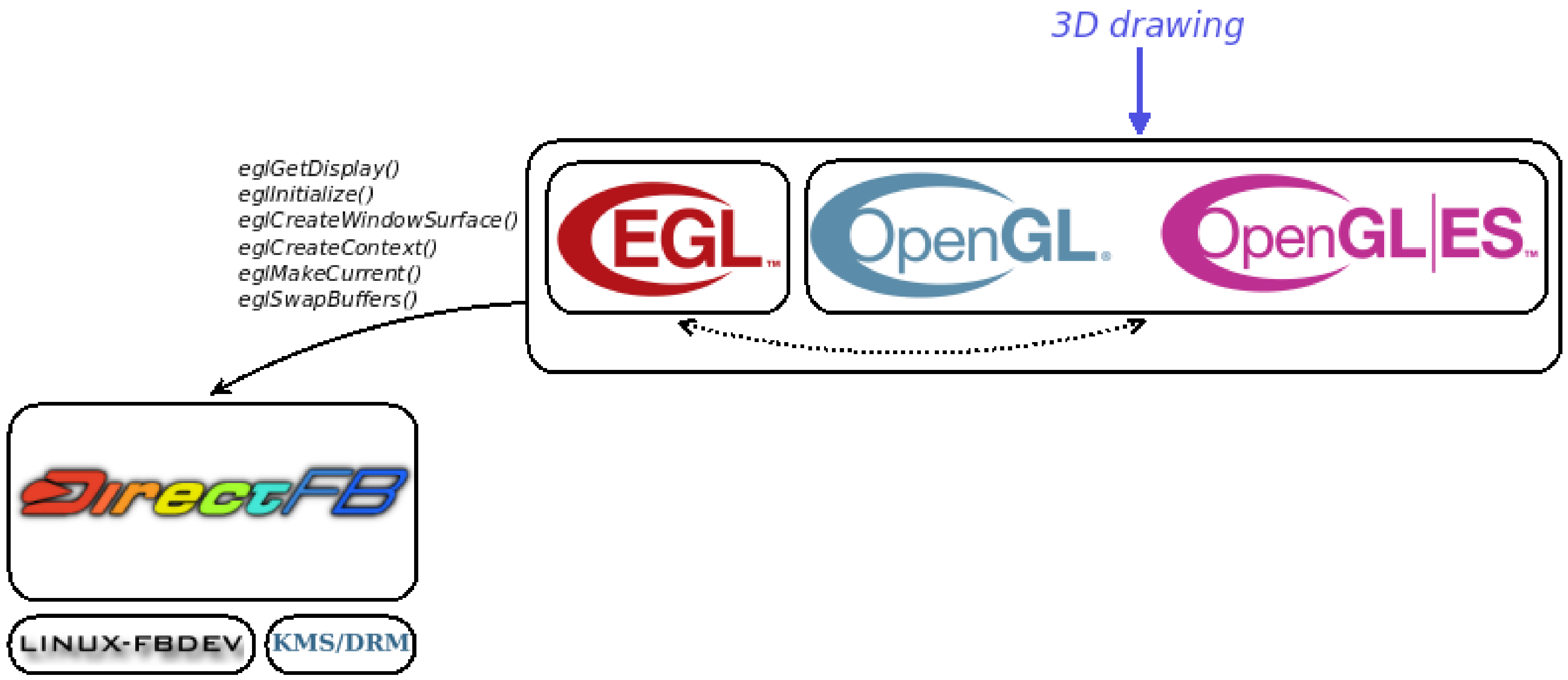
- Implemented in Mesa https://gitlab.freedesktop.org/mesa/mesa
  → DirectFBGL module in *src/mesa/drivers/directfb/idirectfbgl_mesa.c*
- Examples:
  - yagears https://github.com/caramelli/yagears
  - mesa-demos https://gitlab.freedesktop.org/mesa/demos

DirectFBGL interface



```
~$ WIDTH=360 HEIGHT=360 yagears -b gl-directfb -e glesv2 &
~$ df_morph3d --dfb:force-windowed,scaled=480x360
Morph 3D - Shows morphing platonic polyhedra
[1] - Tetrahedron
[2] - Hexahedron
[3] - Octahedron
[4] - Dodecahedron
[5] - Icosahedron
```

EGL interface for DirectFB



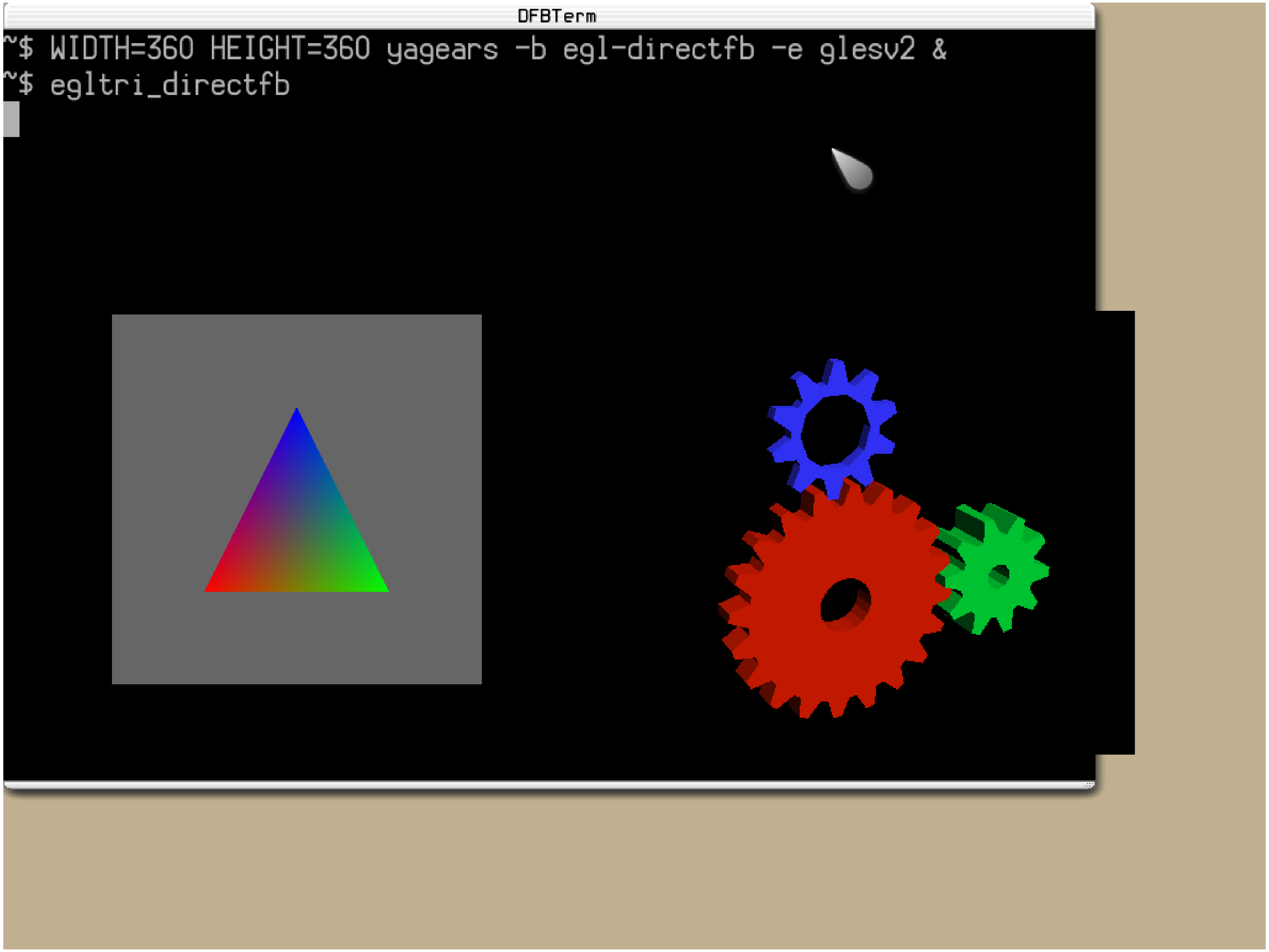
- Implemented in Mesa https://gitlab.freedesktop.org/mesa/mesa

  → DirectFB support in

  src/gallium/state_trackers/egl/directfb/native_directfb.c

  src/gallium/winsys/sw/directfb/directfb_sw_winsys.c
- Examples:
  - yagears https://github.com/caramelli/yagears
  - mesa-demos https://gitlab.freedesktop.org/mesa/demos

EGL interface for DirectFB



```
~$ WIDTH=360 HEIGHT=360 yagears -b egl-directfb -e glesv2 &
~$ egltri_directfb
```

Contents

Vulkan rendering

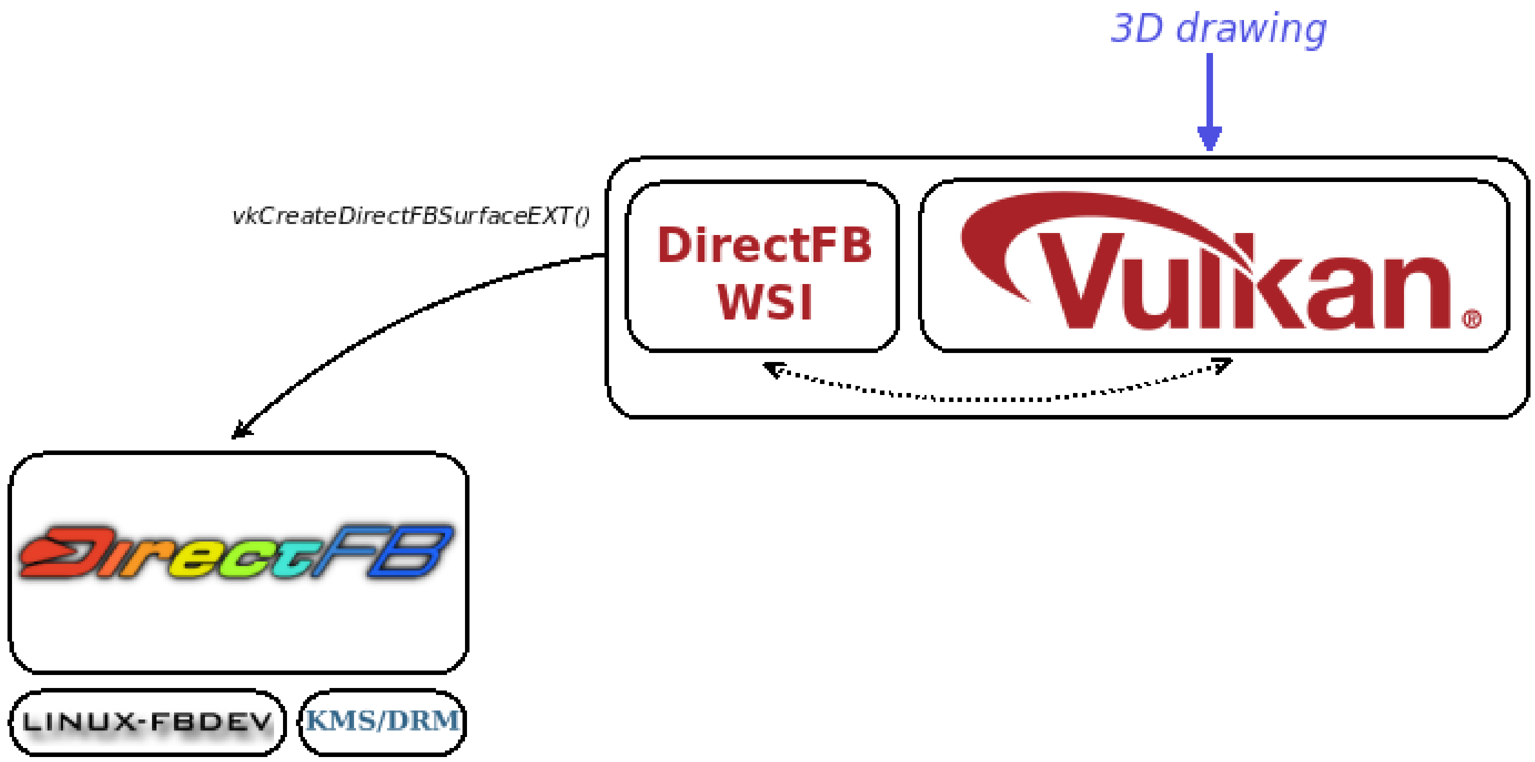
- VK_EXT_directfb_surface extension is used by DirectFB applications for rendering

  → since Vulkan 1.2.146 released in 2020

- The **SwiftShader** project makes Vulkan rendering possible with DirectFB

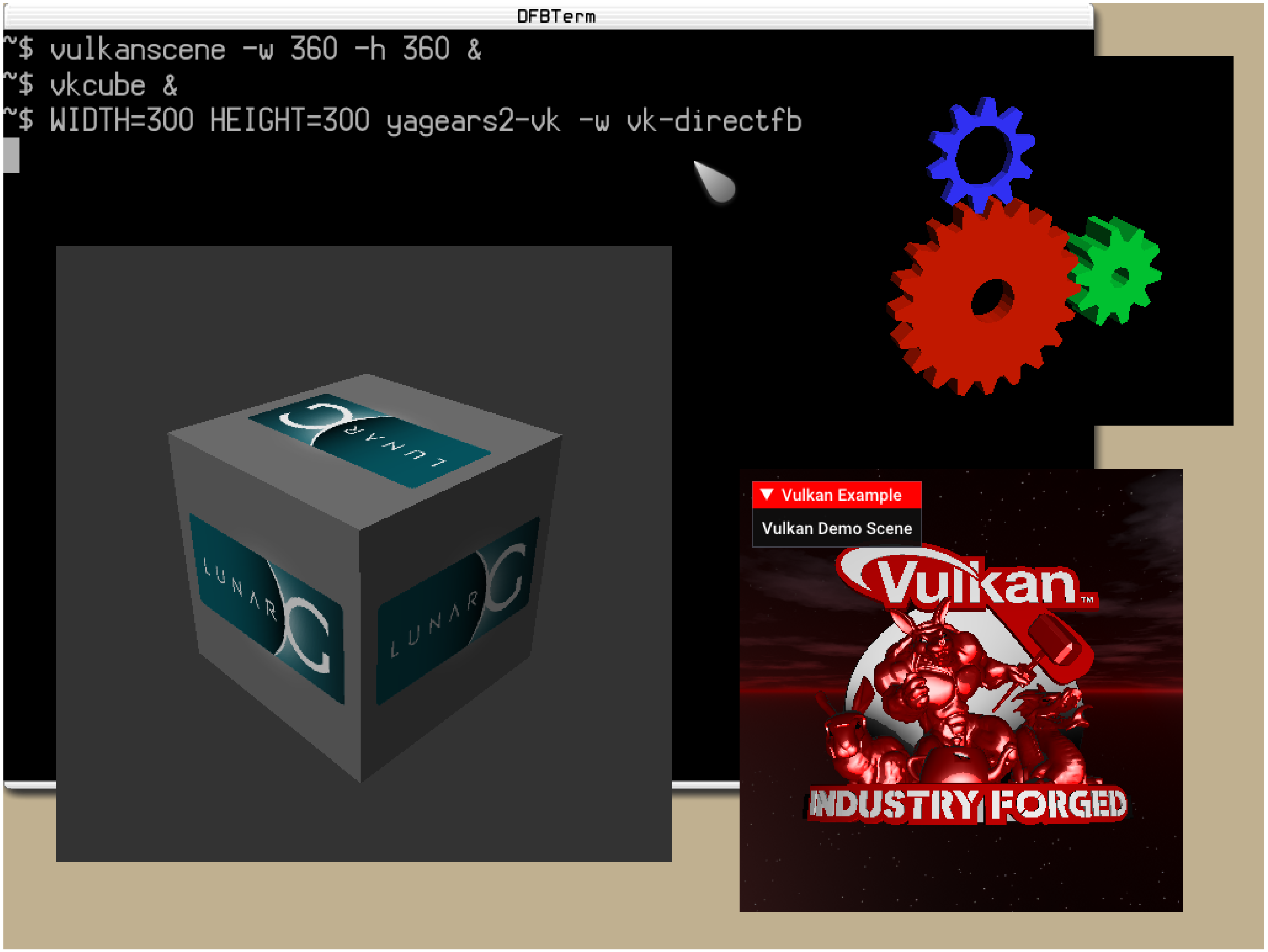
  → mainly used for experimentation and debugging purposes, depending on the chipset, a specific implementation may be available

VK_EXT_directfb_surface extension



- Implemented in SwiftShader https://swiftshader.googlesource.com/SwiftShader
  → DirectFB WSI in *src/WSI/DirectFBSurfaceEXT.cpp*
- Examples:
  - Vulkan-Tools https://github.com/KhronosGroup/Vulkan-Tools
  - Vulkan-Examples https://github.com/SaschaWillems/Vulkan
  - yagears https://github.com/caramelli/yagears

VK_EXT_directfb_surface extension



```
~$ vulkanscene -w 360 -h 360 &
~$ vkcube &
~$ WIDTH=300 HEIGHT=300 yagears2-vk -w vk-directfb
```

And to go beyond with DirectFB...

https://directfb2.github.io

- Programs running directly on DirectFB
  - DFBTerm terminal emulator
  - DFBView image viewer
  - Projektor PDF viewer
  - NetSurf web browser
  - DFBSee media player
- LiTE and ilixi toolkits
- Cairo and Evas drawing libraries
- GLUT and SDL graphics abstraction layers
- GTK+, Qt, Elementary/EFL user interface toolkits
- ...