DELPHIDAY

italian conference

Applicazioni Console con i controfiocchi

Colori, animazioni, widget interattivi nel tuo terminale, senza VCL o FMX: scopriamo le **TUI**!

MARCO BREVEGLIERI

ABLS Team snc - Developer & Trainer

www.compilaquindiva.it

marco.breveglieri@abls.it

twitch.tv/compilaquindiva

youtube.com/@compilaquindiva

linkedin.com/in/marcobreveglieri/

DELPHIDAY

italian conference

wintech italia

OPEN-SOURCE PROJECTS

github.com/**marcobreveglieri**

Prometheus Client Delphi

github.com/marcobreveglieri/prometheus-client-delphi

Murphy for Delphi

github.com/marcobreveglieri/murphy-delphi

Blinki_

github.com/marcobreveglieri/blinki

AGENDA

- Il terminale: ieri e oggi
- TUI (Terminal User Interface): cosa sono?
- Creare TUI con Delphi: è facile!
- Recap & takeaways
- Demo! Demo! Demo!

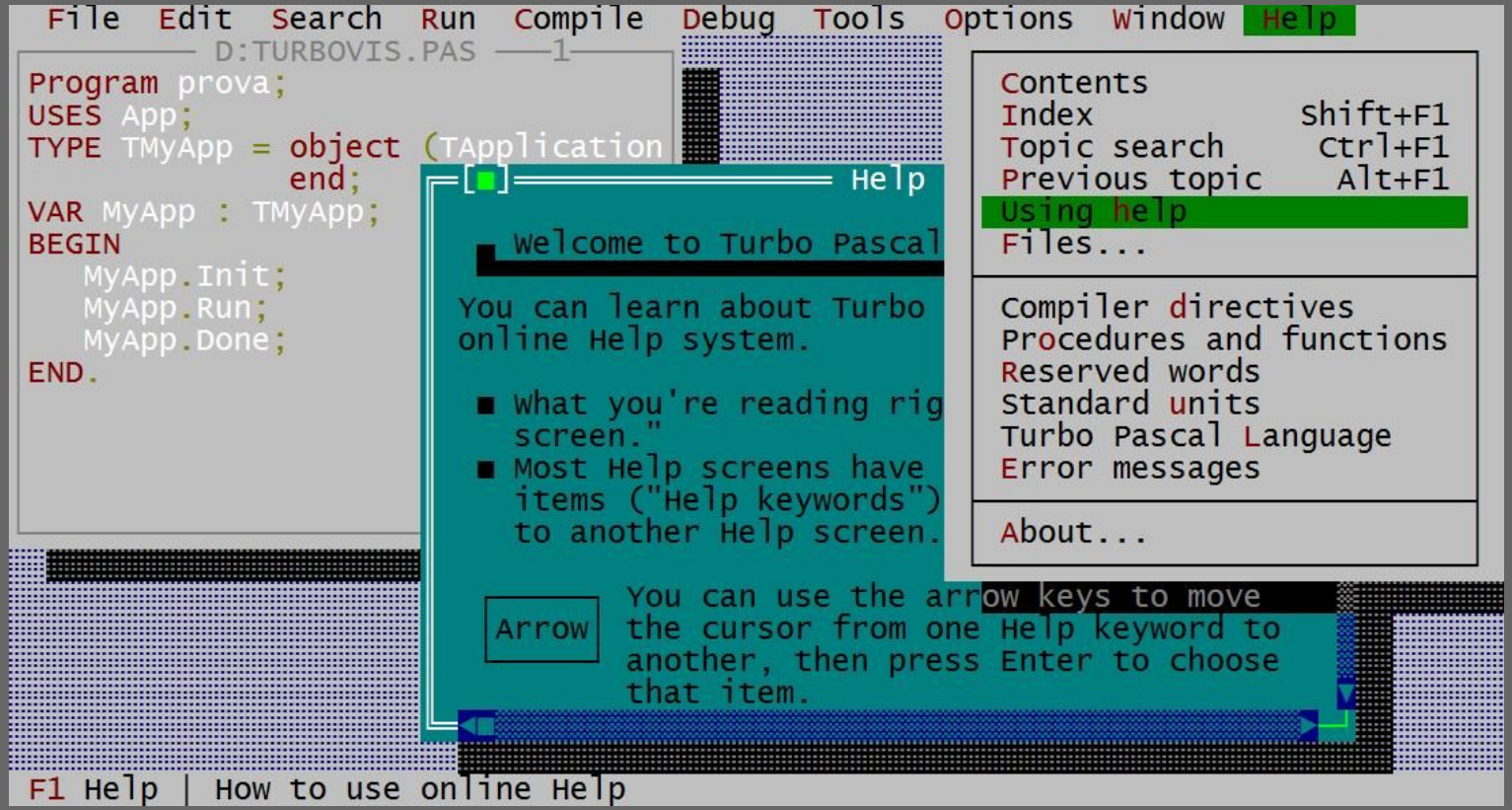
Terminale: ieri e oggi

1

Il “Terminale” (e le sue app)

👉 https://github.com/MrLesk/Backlog.md

Ho avuto un déjà-vu... 🤨



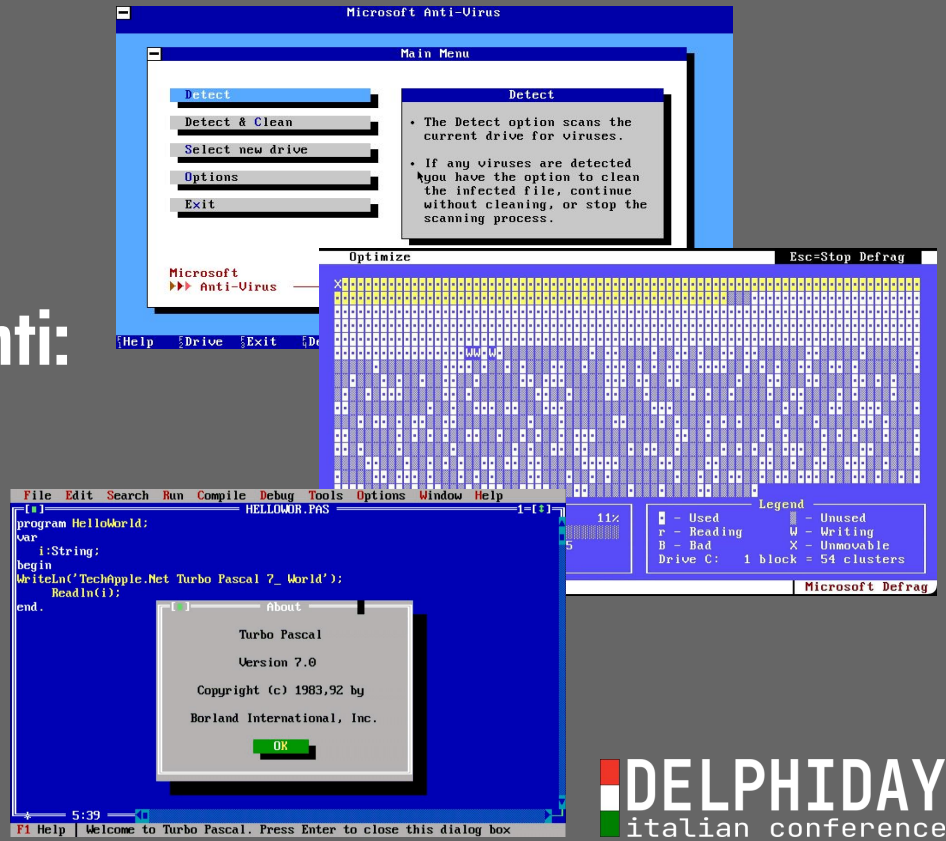
Un po’ di storia...

start cmd.exe

Delphi da sempre supporta applicazioni Console, ma...

Gli anni d'oro del DOS

- **Negli anni '80 e primi '90: il PC è dominato da MS-DOS!**
- **Niente GUI nativa, solo testo**
- **Esistevano "TUI" sorprendenti:**
  - Norton Commander
  - WordPerfect, Lotus 1-2-3, dBase
  - Turbo Pascal IDE



DELPHIDAY italian conference

Turbo Vision

- **E’ uno storico framework per lo sviluppo di interfacce testuali (TUI) organizzate a finestre**
  - Lanciato da Borland nei primi anni ‘90 per MS-DOS
- **Rendeva gli IDE più intuitivi, moderni e facili da usare**
  - Interfaccia innovativa → *menu, finestre, pulsanti, caselle di controllo, persino il supporto per il mouse*
  - Architettura a oggetti → *uno dei primi grandi esempi di librerie OOP con classi riutilizzabili*

Dietro le quinte

- **Manipolazione diretta della memoria video in modalità testo**
  - La scheda video legge costantemente da un indirizzo RAM
  - Ogni singola cella dello schermo sono due byte
    - 1 byte per il carattere, 1 byte per il colore
  - Non si “disegna”: si sovrascrive questi byte (con Z-Order)
- **Usando caratteri speciali si creano bordi, ombre, frecce e cursori**
- **Gestione eventi tastiera e mouse con “Event Loop”**

Il tramonto delle interfacce testuali

- **Le interfacce grafiche (GUI) hanno preso il sopravvento**
  - Non è più possibile accedere direttamente alla memoria
  - Passaggio MS-DOS → Windows 3.x (prima) → Windows 95
  - Developer e utenti volevano app Windows native (e tool)
- **Borland abbandonò lo sviluppo di Turbo Vision**
  - Si concentra su nuove librerie: OWL e una certa VCL… 😉
- **Terminale dici? Solo sysadmin e linuxari…**

Passata la notte... 

La rinascita

- **DevOps e Cloud Native** → Kubernetes, Docker, SSH, Lazygit
- **Revamp dei terminali** → Windows Terminal, iTerm2, Alacritty
  - Uso della GPU, 16 milioni di colori, supporto Unicode
- **Rivoluzione AI** → Claude Code, Copilot CLI, OpenCode
  - Hanno scelto tutti la Console come “mezzo” principale
- **Supporto SSH** → funziona ovunque, zero installazione, sviluppo immediato, sicurezza integrata

Mentre noi sviluppatori continuiamo a scrivere applicazioni VCL, FMX e Console
- e va benissimo -
il resto dei dev ha riscoperto il terminale.

Vogliamo restare a guardare?

TUI (Terminal User Interface)

2

DELPHIDAY
italian conference

Gli ingredienti di una TUI

- **Abilitare la modalità VT della Console** tramite API Windows
- **Emettere sequenze ANSI escape** per colori, posizioni e stili
- **Scrivere caratteri Unicode** per box, blocks, shapes, emoji, ...
- **Mantenere un canvas in memoria** e fare diff prima di scrivere
- **Far girare un event loop** su *ReadConsoleInput()*

1) La modalità VT

Il futuro è nel passato: il DEC VT-100 💖

DEC VT-100 (1978) è il terminale che ha (ri)definito lo standard.

- ➔ 24 righe x 80 colonne per default
- ➔ Capiva le “sequenze ANSI escape”
  - ◆ Sono state codificate poi in ANSI X3.64 ed ECMA-48
  - ◆ Sono in giro da quasi 50 anni: non se ne andranno mai!

DELPHIDAY italian conference

Giochi stile Rogue (roguelike)

Di Artoftransformation - Computer Game Rogue Rendering (TheDraw/Photoshop 4.0), Pubblico dominio.

Abilitare la modalità VT

- Su Windows una chiamata API trasforma la “console legacy” in un terminale VT:

```
GetConsoleMode(hOutput, Mode);
SetConsoleMode(hOutput, Mode
    or ENABLE_VIRTUAL_TERMINAL_PROCESSING
    or DISABLE_NEWLINE_AUTO_RETURN);
```

- Da Windows 10 v1511 (2016): con una riga cambia tutto!
- *Windows Terminal* supporta la modalità VT: usalo!

2) Sequenze ANSI

Sequenze ANSI: cosa sono?

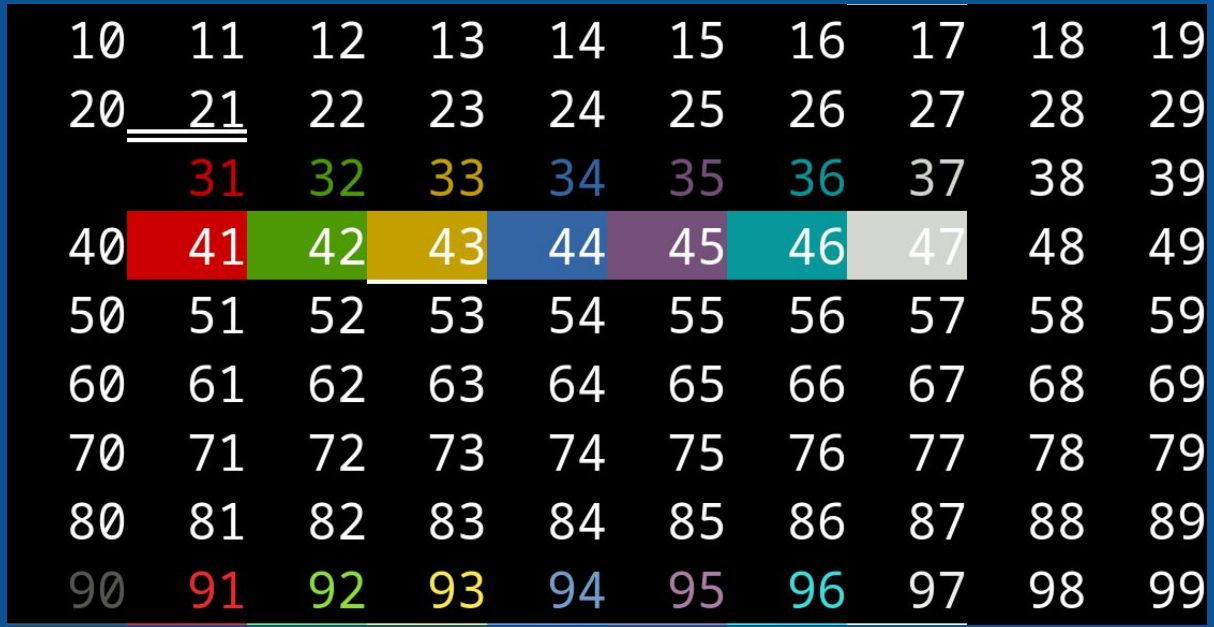
- Una sequenza ANSI è composta da
  `ESC (27) + CSI '[' + parametri + terminatore`

- Alcuni esempi
  `ESC[2J` → *pulisce lo schermo*
  `ESC[31m` → *testo rosso*
  `ESC[10;20H` → *cursore alla riga 10, colonna 20*

Sequenze ANSI: le famiglie

- **SGR (colori e stili)**
  `ESC[1m` bold, `ESC[38;2;R;G;Bm` true-color FG
- **CUP (cursore)**
  `ESC[10;20H` riga 10 colonna 20
- **ED / EL (cancellazioni)**
  `ESC[2J` schermo, `ESC[K` riga
- **Modi speciali**
  `ESC[?1049h` alternate buffer
  `ESC[?25l` nascondi cursore

ANSI



3) Unicode

Caratteri Unicode: i nostri pennelli

- Una TUI non usa pixel, bensì **caratteri**.
- I caratteri **Unicode** ci forniscono gli elementi.

Box Drawing  → *bordi e cornici*

Block Elements  → progress bar, sparkline

Geometric  → indicatori vari

Braille  → animazione spinner

Arrows  → navigazione

4) Canvas + Diff

Performance e flickering

- Se scriviamo direttamente sul terminale, otteniamo **flicker**!
- La soluzione efficace per ogni interfaccia: il **double buffering**
  - Canvas in memoria: matrice WxH di celle (char+FG+BG+attr)
  - A ogni frame, riempio il canvas in memoria
  - Confronto il canvas con la versione precedente (diff)
  - Mando al terminale SOLO le celle cambiate

Senza un algoritmo di “diffing”, una dashboard a 30fps **consumerebbe 10x volte la CPU!**

5) Event Loop

La “centralina” dell’app

- L’input NON si legge ovviamente con *ReadLn()* 😉
- Si usa *ReadConsoleInputW()* che supporta eventi atomici
  - Pressione di tasti, eventi mouse, resize della finestra

```
while not Quit do
begin
  WaitForEvent(Timeout);              // input + tick clock
  DispatchEvent;                      // al widget con focus
  if NeedsLayout then DoLayout;       // measure/arrange
  if NeedsPaint then DoPaint;         // canvas → diff → terminale
end;
```

- *That’s it!* 6 righe coprono il 90% di ogni TUI.

TUI con Delphi

3

DELPHIDAY
italian conference

Footprint ridotto

Piattaforma	RAM tipica	Disco tipica
Electron (VS Code...)	300-600 MB ⚠️	150-300 MB ⚠️
Web + runtime (MAUI self-contained)	100-250 MB	80-200 MB
.NET WPF	60-200 MB	5-80 MB
Delphi VCL/FMX	5-30 MB ⚡	5-25 MB⚡
Delphi TUI (Blinki)	**1-3 MB** 💘	**1-5 MB** 💘

Performance

- **Avvio a freddo rapidissimo (< 50 ms)**
  - (Electron: 1-3 secondi)
- **Prestazioni elevate con basso consumo di risorse**
  - Loop a 30 fps con uso CPU < 1%
- **Gestione ottimale della memoria**
  - Nessun overhead da Garbage Collection (GC)
  - Comportamento prevedibile e deterministico

Stabilità e manutenibilità

Stabilità che dura ormai da 30 anni!

- ➔ Sequenze ANSI stabili dal 1979 (ECMA-48)
- ➔ Unicode è stabile da decenni
- ➔ WinAPI console backward-compatible da NT
- ➔ Object Pascal e Delphi: spesso compila con poche modifiche

Una TUI scritta bene in Delphi oggi sarà realisticamente in produzione (e manutenibile) per tanti anni ancora!

Gli altri linguaggi hanno le loro librerie 😤

- **Rust** — Ratatui (immediate-mode, widget toolkit)
- **Go** — Bubble Tea (Elm Architecture, ~30k★)
- **Python** — Textual (CSS-like) / Rich (output ricco)
- **JS/TS** — Ink (React nel terminale), OpenTUI (fa girare OpenCode)
- **.NET** — Terminal.Gui / Spectre.Console
- **C++** — FTXUI / tvision (port moderno di Turbo Vision)
- **C** — ncurses / notcurses (i classici)

...e Delphi no? 😢

DELPHIDAY
italian conference

Welcome `Blinki`!

Libreria Delphi open-source per la creazione di applicazioni con interfaccia utente terminale (TUI) avanzate.

👉 https://github.com/marcobreveglieri/blinki/

- ➔ Codifica le sequenze di escape ANSI per le esigenze più comuni
- ➔ Comprende un set di widget gerarchici già pronti all'uso

  L'elenco è in rapida espansione… contribuisci con il tuo! 💡
- ➔ Include un renderer a doppio buffer integrato (no flickering)
- ➔ Offre un "event loop" pulito e a singolo thread

ARCHITETTURA DI BLINKI

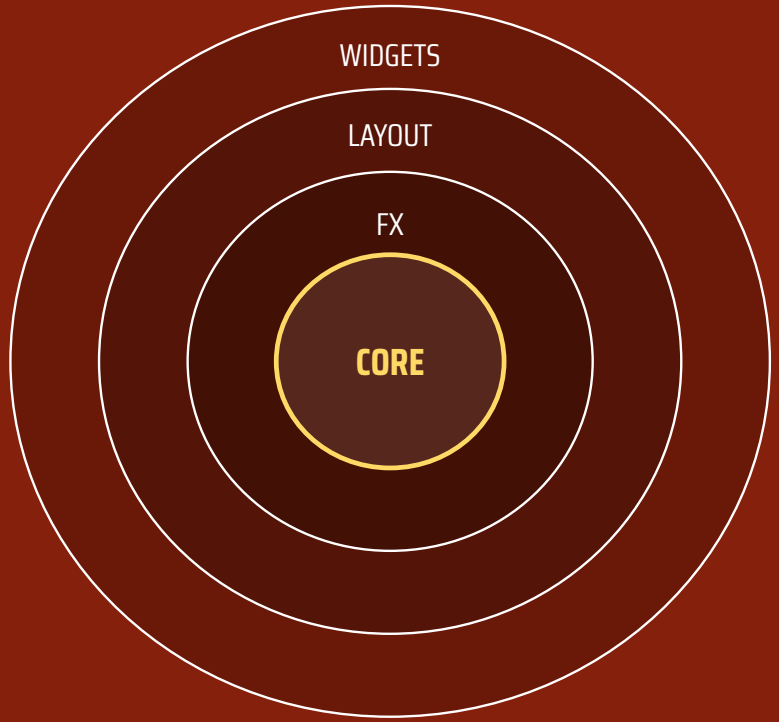
→ Gerarchia a "Cipolla" (4 Layer)

Blinki.Widgets.*
22 widget (Label, Box, Gauge, Table, Menu...)

Blinki.Layout.*
VStack, HStack, Grid, Constraint solver

Blinki.FX.*
Gradienti e altri effetti visivi avanzati

Blinki.Core.*
App, Canvas, Render, Event, Input, Ansi



Ogni layer dipende solo da quelli inferiori.
TTuiWidget è la base: **Measure, Arrange, Paint**.

Sequenze ANSI in Delphi

Posso scrivere un “Hello” di colore rosso e **bold** in Delphi puro?

Certo! In due righe:

```
SetConsoleMode(GetStdHandle(STD_OUTPUT_HANDLE), 7);    // abilita VT
WriteLn(#27'[1;31m', 'Hello!', #27'[0m');               // bold + rosso
```

P.S. “Blinki” fa esattamente questo, solo tramite una API:

```
// ① Livello basso — TTuiAnsi genera le stesse sequenze ESC
var LStyle := TTuiStyle.Create(TTuiColors.Red, {bg} TTuiColor.Default, [taBold]);
Write(TTuiAnsi.ApplyStyle(LStyle) + 'Hello!' + TTuiAnsi.Reset);
// ② Livello canvas — nessuna sequenza manuale, pensa Blinki al diff/flush
Canvas.WriteAt(X, Y, 'Hello!',
  TTuiStyle.Create(TTuiColors.Red, TTuiColor.Default, [taBold]));
```

DELPHIDAY
italian conference

Esempio: Gradient (FX)

- Testo con interpolazione liscia di colori true-color.

```
for i := 1 to Length(s) do
begin
  Color := HSL(ColorStart, ColorEnd, (i-1)/(Length(s)-1));
  Write(Format(#27'[38;2;%d;%d;%dm', [Color.R, Color.G, Color.B]));
  Write(s[i]);
end;
Write(#27'[0m'); // reset alla fine
```

- In **Blinki**:

```
  DrawGradient(ACanvas, LTitleX, ARect.Top, LTitle,
    TTuiColor.RGB(255, 200, 0), TTuiColor.RGB(0, 200, 80),
    Theme.Background, [taBold]);
```

Esempio: Progress Bar (Widget)

- Uso di Unicode per “block elements”
  8 livelli per la cella parziale
  - Celle piene = `Round(Progress × Width);`
  - Cella parziale = ▉ ▊ ▋ ▌ ▍ ▎ ▏

- In **Blinki**:

```
var LProgressBar := TTuiProgressBar.Create( ... );
LProgressBar.Value := 60;
```

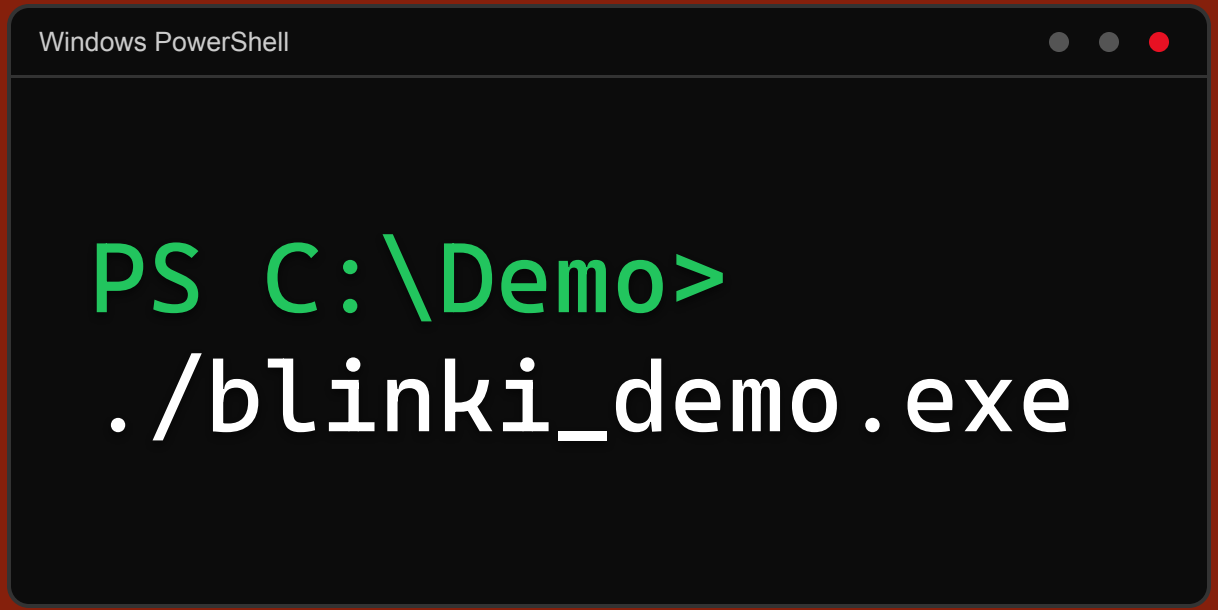
ERGONOMIA DI SVILUPPO

Scrivere TUI in Delphi è piacevole.

- ➔ Object Pascal leggibile per design
- ➔ IDE (autocompletamento, navigazione, ~~refactoring~~)
- ➔ Compilatore velocissimo (build full < 1s)
- ➔ Zero catene npm/pip/ecc o altre dipendenze
- ➔ Debug step-by-step nativo (come per VCL/FMX)

Con Blinki: tanti widget pronti, layout constraint-based, effetti già preconfezionati.

Il programmatore può concentrarsi sul suo problema, non sull'infrastruttura.



DELPHIDAY
italian conference

Concludendo...

4

Recap & Takeaways

- **Le TUI hanno una storia, anche in Pascal.**
  - Turbo Vision (1990) è stato uno dei primi framework TUI completamente OOP del mondo “mainstream”
- **Creare TUI è relativamente semplice e immediato**
  - Basta impostare il VT mode e usare le giuste sequenze ANSI
  - Usa Blinki se ti piace (contribuisci!) oppure crea la tua libreria
- **In molti scenari, una TUI può essere preferibile per tanti motivi**
  - Più semplice, performante, stabile, remotabile
- **Per le TUI, Delphi non è secondo a nessuno, anzi!**

DELPHIDAY
italian conference

What's next?

- **Programma oggi la tua prima TUI!**
- **Usa Blinki e contribuisci al progetto**
  - Provalo, invia le tue PR, issue e "critiche" tramite GitHub
- **Ci sono altri tool interessanti da provare**
  - AR.Console https://github.com/DeerBear/ARConsole
  - Delphi-Terminal https://ideasawakened.com/post/delphi-terminal-a-dockable-console-for-RAD-Studio
  - VSoft.AnsiConsole https://github.com/VSoftTechnologies/VSoft.AnsiConsole

What's next?

➔ Post interessanti

- Costruire applicazioni TUI con Delphi e DMVCFramework
  https://www.danieleteti.it/post/tui-delphi-dmvcframework-it/
- Colori Console alla Colorama (Python) e Log alla Gin (Go) in DMVCFramework
  https://www.danieleteti.it/post/console-colors-dmvcframework-it/

➔ Applicazioni TUI e dove trovarle

- Awesome TUIs https://github.com/rothgar/awesome-tuis
- TUI Tools for Developers https://www.freecodecamp.org/news/essential-cli-tui-tools-for-developers/
- Terminal Apps Dev https://terminal-apps.dev/

Q&A

DELPHIDAY
italian conference


DELPHIDAY
italian conference
> THANK YOU