

[Home](#) > [General Development](#) > The text mode lie: why modern TUIs are a nightmare for accessibility

The text mode lie: why modern TUIs are a nightmare for accessibility

by Thom Holwerda on 2026-05-05 in General Development 12 Comments

There is a persistent misconception among sighted developers: if an application runs in a terminal, it is inherently accessible. The logic assumes that because there are no graphics, no complex DOM, and no WebGL canvases, the content is just raw ASCII text that a screen reader can easily parse.

The reality is different. Most modern Text User Interfaces (TUIs) are often more hostile to accessibility than poorly coded graphical interfaces. The very tools designed to improve the Developer Experience (DX) in the terminal—frameworks like Ink (JS/React), Bubble Tea (Go), or tcell—are actively destroying the experience for blind users.

← [Casey Reeves](#)

The core reason should be obvious: the command-line interface, at its core, is just a stream of data with the newest data at the bottom, linearly going back in time as you go up. Any screen reader can deal with this fairly easily, and while I personally have no need for such a tool, I've heard from those that do that kernel-level screen readers are quite good at what they do. TUIs, or text-based user interfaces, made with modern frameworks are actually very different: they're "2D grid[s] of pixels, where every character cell is a pixel. [They] abandons the temporal flow for a spatial layout."

It should become immediately obvious that screen readers won't really know what to do with this, and Reeves gives countless examples, but the short version is this: the cursor jumps all over the place with every screen update, which makes screen readers go nuts. Various older TUIs, made in a time well before these modern TUI frameworks came about, were designed in a much more terminal-friendly way, or give you options to hide the cursor to solve the problem

that way. [lrssi](#), for example, uses VT100 scrolling regions instead of redrawing the whole screen every time something changes.

I had never really stopped to think about TUIs and screen readers, as is common among us sighted people. The problems Reeves describes seem to stem not so much from TUIs being inherently inaccessible, but from modern frameworks not actually making use of the terminal's core feature set. I really hope this Reeves' article shines a light on this problem, and that the people developing these modern TUIs start taking accessibility more seriously.

About The Author



Thom Holwerda

Follow me on Mastodon [@thomholwerda@exquisite.social](https://thomholwerda@exquisite.social)

12 Comments



2026-05-05 6:32 PM

[sukru](#)

Yes, this should be obvious.

“Graphical” UIs, especially properly designed ones come with pretty neat nested structures with metadata built in.

“Here is a window. At the bottom, there is a bar with two buttons. The button on the left says Okay and has WM_OK action. The one on the right says Cancel and WM_CANCEL action. The title of the ...”

And so on. This is perfectly readable by the screen reader, AI assistant, or whatever your computerized agent is.

The text UI? Requires full on visual understanding. High level machine learning and prone to errors.

Ironic...



2026-05-05 6:36 PM

[RemusRM](#)

is this a rebuttal to this? <https://news.ycombinator.com/item?id=48000028>

I agree, text base is silly these days!

DONATE TO
OSNEWS

YOU CAN ALSO:

- [Become a Patreon](#)
- [Buy OSNews merch](#)



2026-05-05 7:59 PM

[teco.sb](#)

I think the article makes a similar argument that a lot of folks, even here at OSnews, have been making for some time: New interfaces suck! As always, some of these new interfaces are being designed by “first to market”-minded people who throw all past research into user interfaces/interactions out the window. Particularly enlightening, is the section titled Why The “Old Guard” Works. Older applications work well because they were designed for the terminal to solve a specific problem. On the other hand, applications like gemini-cli only exist so some 1337 haxors can feel smug.

For a list of decent terminal utilities that were actually designed for the terminal, check out the Suckless project and some of the projects they recommend (suckless.org/rocks/)



2026-05-05 11:24 PM

[sukru](#)

teco.sb,

They are not for the Terminal, but the Terminal App Requires Unicode, including the smiley graphics. Had all bells and whistles like 256 colors and direct access almost like a framebuffer. Assume extremely fast connection where they can spam output with progress bars.

It would be okay, if left tasteful, and they actually check IS_MODERN_CONSOLE or similar, and have a decent fallback.

But many of them are quick vibe coded experiments, where nobody even read that thousands line shell script.



2026-05-05 9:12 PM

[Shiunbird](#)

Heck... I have perfect eyesight (except colorblind) and this pisses me off too much. Its the same problem like web spinning wheels that burn CPU and convey no messages. Crap like spinning | _ / | to show progress and then if you connect from a serial terminal, you get full garbage and messy lines and unicode emoticons in the terminal.

computers
are
not
toys

2026-05-06 7:17 AM

SEARCH OSNEWS

Search

OSNEWS ACCOUNTS

- [Create Account](#)
- [Login](#)

TOPICS

Select Category

RELATED POSTS

[12 Programming Mistakes to Avoid](#)

December 6, 2010 • [19](#) Comments

[Programming Language Popularity](#)

September 24, 2004 • [83](#) Comments

[Intro to the LLVM MC Project](#)

April 12, 2010 • [One](#) Comment

[LLVM Compiler 2.1 Released](#)



[jurmcc](#)

I can understand your feelings to some degree, but I have to say that for me computers are the best toys I could imagine 😊



2026-05-06 10:46 AM

[Shiunbird](#)

=)))))))))))))

I see your point.

Well, let the toys be toys then, but let the serious tools be serious. I don't really think that emoticons belong in the terminal. TUIs exist for a reason and someone may need to use because they are resource-constrained.

You go, script something, try to collect the output and then you get a bunch of broken characters where the smiles should be, or broken line breaks, and ugh...

Example: I can't use btop. I must use btop -lc -t. Otherwise I get a completely silly text gradient (that reduces readability) and cosmetic characters from the extended set. The defaults should be the clean ones and if I want fancy, then btop -silly-gradient -extended-characters



2026-05-06 5:29 AM

[NaGERST](#)

dvtm is really nice for me.



2026-05-06 6:51 AM

[mbq](#)

LLMs are a pretend game, and pretended coding experience is just more immersive through terminal.



2026-05-06 9:03 AM

[EdgeWiselyInAnnArbor](#)

It is worth noting that the Go TUI "huh" (based on bubbletea/lipgloss) has first-class support for screen readers: <https://github.com/charmbracelet/huh?tab=readme-ov-file#accessibility>



2026-05-06 11:41 AM

[rvss](#)

I believe every TUI based program should have a standard argument based CLI version for which the TUI is merely a

September 29, 2007 • 70

[Comments](#)

[XFS 1.3.0 Released](#)

August 22, 2003 • 28

[Comments](#)

wrapper. That way you can still pipe the command, and likewise, s
can process them for those who use them.



2026-05-13 9:38 PM

[ssokolow \(Hey, OSNews, U2F/WebAuthn is broken on Firefox!\)](#)

A thought that just occurred to me:

“In other words, we’re seeing a TUI version of UI toolkits like Flutter that Chesterton’s Fence their way to throwing out browser built-ins and reinventing them poorly in a canvas tag, bypassing the DOM and all the ambient functionality it brings.”

(At least, with egui and Godot and such, when they just use one big opaque <canvas> element, it’s because they’re unarguably desktop application frameworks and they just happen to support a WebAssembly-based build target for showing off your creations easily. Flutter sort of straddles the line between a mobile UI toolkit and a web UI toolkit.)

POPULAR POSTS

[Codeberg’s programmer user base overwhelmingly votes to ban slopcoded projects from its platform](#)

July 22, 2026 • [82](#)

[Comments](#)

[PervertPods: Apple is adding cameras to AirPods](#)

August 18, 2026 • [48](#)

[Comments](#)

[Happy companies are all alike; every unhappy company is unhappy in its own way](#)

July 22, 2026 • [45](#)

[Comments](#)

[LG monitors silently install software through Windows Update without user consent](#)

July 19, 2026 • [37](#)

[Comments](#)

[Microsoft CVP explains why killing Windows legacy code takes so long](#)

August 6, 2026 • [30](#)

[Comments](#)

POPULAR POSTS (LAST 3 YEARS)

[Is 2024 the year of Windows on the desktop?](#)

June 20, 2024

[Apple intentionally kills web applications for EU users in iOS 17.4 onward to spite its EU users](#)

February 15, 2024

[Made O'Meter helps you easily and quickly avoid American products](#)

March 6, 2025

[Codeberg's programmer user base overwhelmingly votes to ban slopcoded projects from its platform](#)

July 22, 2026

[Some personal news](#)

March 23, 2024

ABOUT OSNEWS

OSnews Privacy Statement

© OSnews Inc. All Rights Reserved.

OSnews and the OSnews logo are trademarks of OSnews.

Theme Customizations by Adam Scheinberg

Proudly powered by Wordpress

Reader comments are owned by the poster.

We are not responsible for them in

IMPORTANT LINKS

- [OSnews Privacy Statement](#)
- [Sponsorship](#)
- [OSnews on Mastodon](#)
- [Notice to Bulk Emailers](#)
- [Contact OSNews](#)
- [Trans rights are human rights](#)

FROM THE ARCHIVES

any way.

All trademarks, icons, and logos shown or mentioned in this web site are the property of their respective owners.

Reproduction of OSnews stories is permitted only with explicit authorization from OSnews.

Reproductions must be properly credited.

Select Month

