IT doupě

# Jak na… Uživatelské rozhraní v Rustu

Konzolová aplikace je sice často nejlepší řešení, ale zkuste ji někdy ukázat vašim rodičům nebo prarodičům. Šance je, že z ní budou mít strach a nebudou vůbec vědět jak ji ovládat nebo se jim vůbec nebude líbit a to i přesto, že na rozdíl od většiny aplikací má terminál v základu tmavý režim. Pokud proto chcete něco, co vypadá profesionálně a intuitivně, uživatelskému rozhraní se zkrátka nevyhnete. A o tom, jak ho použít ve svých Rust aplikacích, budu právě dnes psát.

Všechna okna, která máte na ploše, včetně samotné plochy a tlačítek okolo, se nazývají GUI (Graphic User Interface). Programovacích knihoven kolem něj existuje mnoho, například Qt, GTK, WinUI (Windows) a dokonce i herní enginy jako Unity nebo Godot. Protože používám Gnome, jevilo se mi GTK, které vyvíjejí, jako jasná volba. Proto bych vám ho chtěl ukázat.

Velkou výhodou GTK je, že běží všude, má otevřený kód a jeho nová verze 4 má moderní vzhled. Nevýhodou byly ovšem ovšem až překvapivě malé množství dokumentací. Naštěstí je knihovna dostupná v několika jazycích, takže pokud jste pro ten váš žádné řešení nenašli, stále můžete zkusit přeložit nějaké jiné. Pro naše účely doporučuji tuto stránku: <https://gtk-rs.org/gtk4-rs/git/book/>.

Zkusíme napsat aplikaci s tlačítkem, které po stisknutí přičte +1. Na začátek importujeme **gtk4** knihovnu. Chvíli to potrvá, to je bohužel jedna z mála nevýhod carga. Do main() funkce přidáme proměnnou **app**, která bude sloužit jako základ celé aplikace. Ta vyžaduje id, což může být úplně cokoliv, odděleno třemi tečkami. Teď potřebujeme nějaké GUI. K aplikaci ho připojíme pomocí **app.connect\_activate()**, která celé uživatelské rozhraní, vložené jako parametr, načte. Funkci zakončíme příkazem **app.run()**, který následně aplikaci pustí.

Všechny elementy, se kterými budeme pracovat, se jmenují widgety a mají mnoho podob, od tlačítka, obrázku, jmenovky, progress baru až po kalendář nebo color picker. Všechny si můžete prohlédnout zde: <https://docs.gtk.org/gtk4/visual_index.html>, nás zatím budou zajímat jen 3 – okno, mřížka a tlačítko. Všechno budeme psát do metody **build\_ui()**, protože je již napojená na aplikaci. První uděláme okno. Každý prvek můžeme pomocí **NázevPrvku::builder()** jednoduše poskládat, jako například u okna nastavit aplikaci (ta je v parametru naší funkce), výšku, šířku a title, se kterým budeme pracovat. Nezapomeňte na **window.show()**, jinak se vám nic neukáže!

Abychom mohli do okna přidávat elementy, použijeme grid. I ten má své parametry, které mu přes buildera můžeme nastavit, nás bude zajímat margin, horizontální a vertikální zarovnání a taky rozestupy řádků a sloupců… pokud jste si ještě nezotročili žádného front-end developera, teď je na to ideální čas. A i tady je něco, na co byste neměli zapomenout, a to konkrétně připojit pomocí **window.set\_child()** grid k samotnému oknu.

A už zbývá jen tlačítko, tady bude projednou i trochu logiky. Tentokrát ani nepotřebujete buildera používat, stačí vám pomocí **with\_label()** nastavit text. Abysme na to náhodou nezapomněli, přišpendlíme ho ihned funkcí **grid.attach()** do gridu. Protože kvůli magii Rustu nemůžeme jednoduše hýbat proměnnými z funkce do funkce, uděláme novou proměnnou s kopií okna a tu vložíme do eventu **button.connect\_clicked**, který se spustí, když na tlačítko klikneme (kdo by to čekal). Kam vypíšeme výstup po kliknutí je na nás, já si na ukázku vybral horní panel. Takže teď už jen stačí vzít název okna, přeparsovat ho na int, přičíst k němu 1, nastavit ho jako nový název a máme oficiálně hotovo.

Tím bychom měli základní aplikaci hotovou. Jak vidíte, na to, že skoro nic neumí, má 40 řádků. Proto doporučuji oddělit si alespoň GUI od veškeré logiky, jinak se v tom za chvíli nevyznáte. Toho se dá docílit i přepsáním tohodle všeho do XML podoby, která je kratší a nebude vám nikam zasahovat. Přeji vám s GUI hodně štěstí, pevné nervy při stylování (budete je potřebovat) a u dalšího čísla na viděnou.

//img1,2